International Journal of Robotics and Control Systems
Vol. 5, No. 6, 2025, pp. 2957-2994
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs

A Nonlinear Automatic Control Strategy for Autopiloting a Fourth-Generation Combat Aircraft: The Soukhoï Su-30

Belkacem Bekhiti [a,1], Kamel Hariche [b,2], Alfian Ma'arif [c,3], Abdel-Nasser Sharkawy [d,e,4,*]

[a] Institute of Aeronautics and Space Studies, (IASS), University of Blida 1, BP 270 Blida (09000), Algeria
[b] Institute of Electrical and Electronic Engineering, M'hamed Bougara University Boumerdes, Boumerdes 35000, Algeria
[c] Department of Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
[d] Mechanical Engineering Department, Faculty of Engineering, Qena University, Qena 83523, Egypt
[e] Mechanical Engineering Department, College of Engineering, Fahad Bin Sultan University, Tabuk 47721, Saudi Arabia
[1] belkacem1988@hotmail.co.uk; [2] khariche@yahoo.com; [3] alfian.maarif@te.uad.ac.id;
[4] abdelnassersharkawy@eng.svu.edu.eg
* Corresponding Author

ARTICLE INFO

Article history
Received August 26, 2025
Revised October 10, 2025
Accepted December 11, 2025

Keywords
Su-30 Fighter Aircraft;
Adaptive Dynamic Inversion;
Neural Flight Control;
Pure-Pursuit Guidance;
Multivariable Control Systems

ABSTRACT

This paper proposes a new nonlinear control and guidance strategy for fourth-generation fighter aircraft, demonstrated on the Soukhoï Su-30. The method integrates symmetric and asymmetric control structures with a sliding-mode-enhanced pure-pursuit guidance law within a unified nonlinear dynamic inversion (NDI) framework. This represents the first implementation of such an integrated design on the Su-30 platform. An adaptive radial-basis-function neural network compensates for modeling uncertainties, while aerodynamic parameters are identified using the total least squares estimation method. The approach is validated through Software-in-the-Loop simulations in the SP-ACS environment linked to Microsoft Flight Simulator 2004. Results confirm precise target tracking, smooth maneuvering, and strong robustness to un-modeled dynamics, demonstrating its suitability for advanced combat operations.

1. Introduction

The demand for super-maneuverability in modern and future fighter aircraft continues to grow. Interest in post-stall maneuvers surged after the Su-27's Cobra display at the 1989 Paris Air Show, which highlighted extraordinary post-stall control. Since then, such maneuvers have become a benchmark for evaluating fighter performance [1]. These aggressive maneuvers, occurring at the edges of the flight envelope, involve strong nonlinearities, unsteady dynamics, and complex couplings between aerodynamics, control, and structural responses [2]. With expanded flight envelopes, aircraft like the Su-30, MiG-35, F-22, and F-35 operate effectively at high angles of attack and sideslip. However, these regimes introduce un-modeled dynamics and parameter variations due to unsteady aerodynamics, actuator constraints, and intensified motion coupling [3], [4]. Ensuring control under such conditions (especially during faults or structural damage) requires advanced nonlinear adaptive control capable of handling uncertainty and variability. With the rise of close-range combat, maneuvers like evasion, pursuit, and nose-pointing became critical for first-shot opportunities, driving the need for high-angle-of-attack flight and prompting advanced

maneuverability research [5]. Fighter control systems aim to produce rapid translational and angular accelerations, often requiring large aerodynamic surfaces to generate sufficient forces and moments [6]. The NDI, especially when combined with neural network-based adaptation, has proven effective for this challenge [7]. Commonly used in robotics and vehicle control, NDI is also referred to as computed torque control, force control, or feedback linearization, and is widely studied in control literature [8]. In flight control, it enables consistent performance across the full envelope without gain scheduling by exploiting aircraft nonlinearities [9]-[11]. Key advantages include decoupling longitudinal and lateral dynamics at high angles of attack [12], independent closed-loop dynamics assignment [13], and simplified controller design for each channel [14].

The Su-30MKA "Flanker-C," customized for Algeria, integrates Algerian, French, and Israeli avionics, and shares core features with the Su-35. The first Russian-made Su-30 was delivered to the IAF in 2002, and the first Algerian-assembled unit entered service in November 2006. With around 63 in service, it forms the backbone of the Algerian Air Force [15]. Its primary flight control surfaces include flaperons (δ_{AR}, δ_{AL}) for roll, twin rudders (δ_{RR}, δ_{RL}) for yaw, and horizontal tailplanes (δ_{HR}, δ_{HL}) for pitch. The flaperons act as both ailerons and flaps. Unlike trainers such as the L-39 Albatros, the Su-30 is highly sensitive to inputs, requiring minimal deflection [16]. In flight, differential flaperon deflection (from stick inputs) governs roll, while in-phase deflection (from automatic signals) increases lift based on angle of attack. During takeoff and landing, flaperons deflect together as flaps, while roll remains controlled through differential deflection [17], [18]. The Su-30 combines aerodynamic refinement, thrust vectoring, and fly-by-wire control to deliver exceptional maneuverability and strong takeoff/landing capabilities. It performs extreme maneuvers like the Pugachev Cobra and tail slide, intended to disrupt enemy tracking, though their combat utility remains unconfirmed [19], [20].

The autopilot covers the full flight envelope, supporting terrain-following, multi-target engagement, waypoint tracking, target approach, return, and automatic landing [21], [22]. Built on electronic stability, the Su-30 lacks mechanical controls and depends entirely on fly-by-wire systems [23], [24]. Advanced model-based strategies are used to handle its nonlinear, uncertain dynamics, including NDI [25]-[27], incremental nonlinear control [28], [29], geometric methods [30], [31], adaptive and backstepping control [32]-[35], model reference adaptive control (MRAC) [36], [37], and variable structure control [33]. These avoid gain scheduling, adapt to changing conditions, and support nonstandard regimes. Hybrid approaches, such as NDI with neural networks (NN), have shown success in [38], [39]. Recent advances in distributed and cooperative control, including radial basis neural networks (RBFNN), NN-based target enclosing for multi-agent systems, formation tracking over signed networks, and secure localization under adversarial conditions, offer valuable insights for extending autonomous guidance to multi-vehicle scenarios. While this work focuses on a single combat aircraft (no thrust-vectoring), these methodologies form a strong foundation for future research in networked combat control.

Conventional nonlinear guidance and control methods such as proportional navigation (PN), augmented pursuit (AP), and standard nonlinear dynamic inversion (NDI) have been effective in missile and aircraft control. However, these approaches often rely on simplified models, constant feedback gains, or affine dynamics. PN is purely kinematic and cannot adapt to maneuvering targets. AP improves interception but assumes known dynamics and a separated guidance–control structure. NDI depends on precise modeling and its performance deteriorates under system uncertainties or under-actuation. Compared with previous works that separately combined NDI with neural adaptation [7], [25], [39] or applied pursuit or proportional-navigation laws in isolation [30], the present study introduces a unified hybrid architecture in which *symmetric* and *asymmetric* control loops are jointly optimized and coupled with a *sliding-mode-enhanced pure-pursuit guidance law*. This integration (NDI + RBFNN + pursuit law strategy) allows simultaneous regulation of attitude, speed, and directional guidance within one adaptive framework, extending beyond the decoupled or constant-gain designs typical of earlier (NDI + RBFNN) or (AP + PN) methods. The main contributions of this work can be summarized in:

- Adaptive nonlinear control architecture is developed for conventionally configured fighter aircraft (symmetric control). To address system uncertainties and un-modeled nonlinear dynamics, the design integrates an adaptive mechanism based on neural networks employing radial basis functions (RBFs).
- Since symmetric control does not serve all the purpose, because the controlled system is under-actuated (that is the four inputs can control only four outputs) we proposed a new intelligent differential control strategy. This method is able to control both position and angular steering guidance.
- Finally, simulation/validation of the new *practical adaptive autopilot* are carried out through a SP-ACS, which is an environment framework based on "Software In the Loop (SIL) methodology" and we use Microsoft Flight Simulator (FS2004) as the environment. FS2004 has a rich library of aircraft. This library is extensible by adding and customizing airplanes, and flight instruments.

This approach improves control performance under uncertainty, minimizes dependence on precise aerodynamic models, and proves effective in evasive maneuvers. It uniquely combines symmetric and asymmetric control strategies for combat guidance, enabling smooth transitions between control modes.

The paper is organized as follows: Section 2 outlines the aircraft model and problem formulation. Section 3 presents the symmetric control design based on a 6DOF nonlinear Su-30MKA model. Section 4 introduces an asymmetric control scheme with neural network-based adaptation. Section 5 describes a practical adaptive strategy. Section 6 shows numerical simulations validating the method's effectiveness, followed by conclusions.

2. Aircraft Description and Problem Formulation

2.1. Aircraft Description

The Sukhoi Su-30MKA (Fig. 1) is a twin-engine, multirole combat aircraft co-developed by the Sukhoi Design Bureau and the Algerian Air Force. It integrates French and Russian avionics, thrust vectoring, and canards. The control inputs include thrust command (δ_{TH}) and deflections of elevators (δ_E for pitch), ailerons (δ_A for roll), and rudders (δ_R for yaw). Aircraft parameters such as wing area s, wingspan b, mean aerodynamic chord c, and mass m are constant, with values listed in Table 1.

Table 1. Parameters and tech data of Su-30MKA

Parameters	Nominal values	Parameters	Nominal values
Length (Tail to Nose)	ℓ =21.935 m	Range (high altitude)	R = 3,000 km
Wingspan (Full Width)	b = 14.7 m	Service ceiling	17,300 m
Wing root chord	c = 3.48 m	Rate of climb	230 m/s
Height	h_{aircraft} = 6.36 m	Wing loading (full fuel)	468.3 kg/m^2
Wing surface area	s = 62 m^2	Thrust/weight	0.86
Empty weight	m_0 = 17,700 kg	Pitch Moment of Inertia	I_{yy} = 230,000 kg. m^2
Gross weight	m_c = 24,900 kg	Roll Moment of Inertia	I_{xx} = 29,000 kg. m^2
Max takeoff weight	m_{max} = 34,500 kg	Yaw Moment of Inertia	I_{zz} = 270,000 kg. m^2
Fuel capacity	m_{Fuel} = 9,400 kg	Cross Moment of Inertia	I_{xy}, I_{xz}, I_{yz} = 0 kg. m^2
Max speed (high R)	V_{max} = 2,120 km/h		

The Sukhoi Su-30MKA (Fig. 1) has more accurate navigation system, a TV command guidance system, a guidance system for anti-radiation missiles, a larger monochrome TV display system in rear cockpit for ASM guidance, and ability to carry one or two pods, typically for laser designation or ARM guidance in association with Pastel RWR and APK-9 datalink. Western avionics, guidance pods and weapons can be fitted optionally.

The Su-30MKA is a highly versatile 4th generation multirole combat aircraft, designed for a wide range of tactical operations. Table 2 outlines its main structural components and onboard systems, as well as the variety of missions it is capable of executing, highlighting its strategic value in modern aerial warfare.

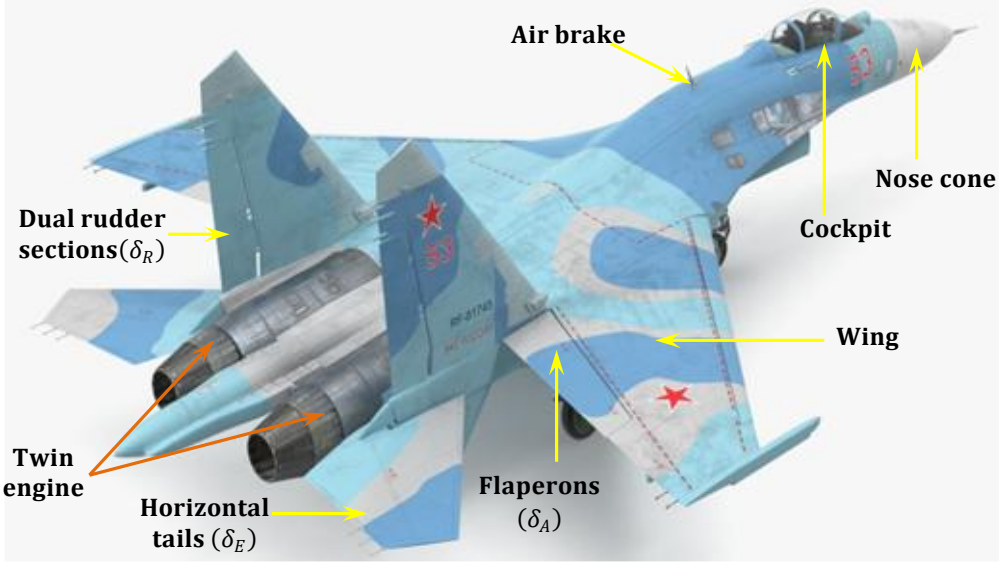

Fig. 1. Sukhoi Su-30MKA (Russian-Algerian combat aircraft)

Table 2. Structural components and mission capabilities of the Su-30MKA multirole combat aircraft

Components and System Construction
The Su-30MKA is made of the following components: • Wings, fuselage, tail assembly, • Propulsion system (Twin AL-31FP turbofan engines) • Thrust vectoring nozzles (i.e. in all directions) • Control surfaces (elevators-ailerons-rudders-canards) • Fuel system (Includes multiple internal fuel tanks) • Flight controller, (digital fly-by-wire system) • Autopilot system (automatic flight control/navigation) • Inertial navigation System and GPS Module, • Telemetry system, track diff-parameters during flight • Payload (e.g., weapons, bombs, sensors and cameras) • Landing gear, tricycle retractable landing gears. • Communication system, radio, data link/SATCOM • Auxiliary power unit (run when the engines are off)
Operations and Missions
Su-30 can perform many operations and missions: • Air Superiority: engage enemy aircraft/control the sky • Ground Attack: precision strikes on ground targets. • Reconnaissance: through surveillance missions. • Electronic Warfare: disrupting enemy systems. • Maritime Strike: destroying naval (anti-ship missiles). • Escort: Protecting other aircraft, such as bombers etc • Close Air Support: Aiding ground forces. • Conducting intelligence operations on the battlefield This versatility allows the aircraft to adapt to various mission profiles, making it a crucial asset in modern air forces.

The Su-30MK is a Russian-made (It gives Algerian Air Force the permission of maintenance, assembly and modification of Su-30, see Fig. 2) fighter aircraft that is distinguished by its ability to reach very high-level and it performs a superior combat maneuver (Cobra, Split-S, Jinking, Lag/lead Pursuit etc).

Fig. 2. Sukhoi Su-30MK aircraft assembly and modification center

2.2. Problem Formulation

The Su-30 aircraft represents a complex nonlinear dynamic system where precise control is challenged by modeling uncertainties, un-modeled dynamics, and actuator constraints. This study addresses three core problems, which are stated:

- **Problem1:** the first being *symmetric control* for robust path tracking and speed regulation. A hierarchical multi-loop control strategy is developed: the inner loop (fast mode) regulates angular dynamics, while the outer loop (slow mode) governs position and orientation tracking. The system is modeled as a non-affine nonlinear dynamics:

$$\dot{x}_1 = f_{sym1}(x_1, x_2, \delta_{TH}) + g_{sym1}(x_1, x_2)u + d_1 \text{ and } \dot{x}_2 = f_{sym2}(x_1, x_2, \delta_{TH}) + d_2 \quad (1)$$

Here, $u = [\delta_A \ \ \delta_R \ \ \delta_E]^T \in \Re^3$ is the aerodynamic control input vector (aileron, rudder, elevator), and δ_{TH} is the thrust input. The state vectors are $x_1^T = [v^T \ \ \omega^T] \in \Re^6$ (body-frame velocities) and $x_2^T = [r^T \ \ \vartheta^T] \in \Re^6$ (inertial position and orientation). Disturbances d_1, d_2 and their derivatives are assumed bounded: $\|d^\nu d_i/dt^\nu\| \le \bar{d}_i \ \ (i = 1.2, \ \nu = 0{,}1)$. To handle such uncertainties, the model is reformulated using neural approximations as.

$$\begin{cases} \dot{V}(t) = f_V^{sym}(x) + f_{1u}^{sym}(x)u + f_{1th}^{sym}(x)\delta_{TH} + d_V \\ \dot{\omega}(t) = f_\omega^{sym}(x) + f_{2u}^{sym}(x)u + f_{2th}^{sym}(x)\delta_{TH} + d_\omega \\ \dot{\vartheta} = f_\vartheta^{sym}(x) + + d_\vartheta \end{cases} \quad (2)$$

where d_V, d_ω, d_ϑ are the un-modeled dynamics and terms accounting for disturbances. The control law is constructed as a nested structure of neural-adaptive subsystems:

$$\begin{cases} u(k) = \mathcal{C}_1\big(\omega(k), \omega_c(k), \delta_{TH}(k), d_\omega(k)\big) & \text{fast\_mode controller} \\ \delta_{TH}(k) = \mathcal{C}_3\big(x(k), V_d(k), u(k-1), d_V(k)\big) & \text{speed controller} \\ \omega_c(k) = \mathcal{C}_2\big(\vartheta(k), \vartheta_d(k), d_\vartheta(k)\big) & \text{slow\_mode controller} \\ \vartheta_d(k) = \mathcal{C}_{GP}\big(r(k), r_d(k), V_d(k)\big) & \text{guidance processor} \\ d_V(k) = NN\big(x(k), V_d(k)\big); \ \ d_\omega(k) = NN\big(x(k), \omega_c(k)\big); \ \ d_\vartheta(k) = NN\big(x(k), \vartheta_d(k)\big) \end{cases} \quad (3)$$

where $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$ is three diff-nonlinear controllers, and $\mathcal{C}_{GP}$ is the guidance processor. $\omega(k)$ is the angular velocity of the aircraft in the body-frame. $\omega_c(k)$ is the commanded angular velocity of the aircraft. $\vartheta(k)$ is the Euler angles (body relative to inertial frame). $\vartheta_d(k)$ is the (indirect) desired

relative rotations of the aircraft. $\mathrm{r_d}(k)$ is the desired position vector in the inertial frame. $\mathrm{V_d}(k)$ is the desired speed of the aircraft. The NNs are designed such that: $\mathrm{NN}(\mathrm{x}, V_\mathrm{d}) = \hat{\mathrm{d}}_V(k) + \mathrm{e}_V(k)$, $\mathrm{NN}(\mathrm{x}, \omega_\mathrm{c}) = \hat{\mathrm{d}}_\omega(k) + \mathrm{e}_\omega(k)$ and $\mathrm{NN}\big(\mathrm{x}(k), \vartheta_\mathrm{d}(k)\big) = \hat{\mathrm{d}}_\vartheta(k) + \mathrm{e}_\vartheta(k)$, where the terms $\mathrm{e}_V(k)$, $\mathrm{e}_\omega(k)$, and $\mathrm{e}_\vartheta(k)$ represent approximation errors. The control objective is to design an advanced guidance and control law for the Su-30 such that the tracking error $\mathrm{e} \to 0$, i.e., remains bounded.

- **Goal 1:** (a) The control task is given as: design an advanced guidance and control law for Su-30 under the constraints, $\sup_{t\in[t_0,\infty]} \|\mathrm{e_r}(t)\| \le \epsilon$; $\sup_{t\in[t_0,\infty]} \|\mathrm{e}_\omega(t)\| \le \epsilon$; $\sup_{t\in[t_0,\infty]} \|\mathrm{e}_\vartheta(t)\| \le \epsilon$; and $\sup_{t\in[t_0,\infty]} \|e_\mathrm{V}(t)\| \le \epsilon$, where the tolerance error $\epsilon > 0$ is a desired arbitrarily small bound between the controlled outputs and its references, the errors are $\mathrm{e_r}(t) = \mathrm{r_d}(t) - \mathrm{r}(t) \in \Re^3$, $\mathrm{e}_\omega(t) = \omega_\mathrm{c}(t) - \omega(t) \in \Re^3$, $\mathrm{e}_\vartheta(t) = \vartheta_\mathrm{d}(t) - \vartheta(t) \in \Re^3$, and $e_\mathrm{V}(t) = \mathrm{V}_d(t) - \mathrm{V}(t) \in \Re^1$. (b) Due to the physical limitations on actuators, thus there exists bounded set, such that $|\mathrm{u}(k)| \le \mathrm{u_{max}}$ and $|\delta_{TH}(k)| \le (\delta_{TH})_\mathrm{max}$. Based on this symmetric configuration and the above assumption, the tracking errors for guidance and attitude dynamics can all converge to a zero within a short time.
- **Problem2:** Conventional symmetric surface control fails to ensure robust path tracking and precise attitude regulation in under-actuated, highly dynamic systems, where limited inputs govern multiple outputs. Moreover, attitude is often treated as a secondary effect rather than a primary goal. To address this, we propose a differential (*asymmetric surface*) control strategy, reformulating the system as:

$$\dot{\mathrm{x}}_1 = \mathrm{f_{asy1}}(\mathrm{x}_1, \mathrm{x}_2) + \mathrm{g_{asy1}}(\mathrm{x}_1, \mathrm{x}_2)\delta_\mathrm{asy} + \mathrm{d}_1 \quad \text{and} \quad \dot{\mathrm{x}}_2 = \mathrm{f_{asy2}}(\mathrm{x}_1, \mathrm{x}_2) + \mathrm{d}_2 \tag{4}$$

where $\delta_\mathrm{asy} = [\delta_{LA}\ \delta_{RA}\ \delta_R\ \ \delta_{LE}\ \ \delta_{RE}\ \ \delta_{TH}]^T \in \Re^6$, and the state are $\mathrm{x}_1^T = [\mathrm{v}^T\ \ \omega^T] \in \Re^6$and $\mathrm{x}_2^T = [\mathrm{r}^T\ \vartheta^T] \in \Re^6$. $\mathrm{v} = [u\ \ v\ \ w]^T$ the linear velocity in the bod-fixed-frame. $\omega = [p\ \ q\ \ r]^T$ is the angular velocity vector in the bod-frame. $\mathrm{r} = [x\ \ y\ \ z]^T$ is the linear position vector in the inertial-frame. $\vartheta = [\phi\ \theta\ \psi]^T$ is the angles of body relative to inertial frame. Given these challenges, the objective is to develop a robust autopilot design that not only ensures desired path following and velocity control but also adapts to changing flight orientations and system dynamics.

$$\left\{\begin{array}{ll} \delta_\mathrm{asy}(k) = \mathcal{C}_1\big(\mathrm{x}_1(k), \mathrm{x}_2(k), \mathrm{x}_{1c}(k), \mathrm{d}_1(k)\big) & \text{inner loop} \\ \mathrm{x}_{1c}(k) = \mathcal{C}_2\big(\mathrm{x}_1(k), \mathrm{x}_2(k), \mathrm{x}_{2d}(k), \mathrm{d}_2(k)\big) & \text{outer loop} \\ \mathrm{d}_1(k) = \mathrm{NN}\big(\mathrm{x}_1(k), \mathrm{x}_2(k), \mathrm{x}_{1c}(k)\big); \quad \mathrm{d}_2(k) = \mathrm{NN}\big(\mathrm{x}_1(k), \mathrm{x}_2(k), \mathrm{x}_{2d}(k)\big) & \end{array}\right\} \tag{5}$$

where x_{1c} is the commanded state vector and $\mathrm{x}_{2d} = \big[\mathrm{r}_d^T\ \ \vartheta_d^T\big]^T$ is the desired state vector.

- **Goal 2:** The asymmetric surface control task can be given as: design an advanced guidance and control law such that $\forall\, t > t_0$, $\|\mathrm{x}_2(t) - \mathrm{x}_{2d}(t)\| \le K\|\mathrm{x}_{2d}(t_0)\|e^{-\lambda(t-t_0)}$ in small ball $\mathcal{B}$ around the origin, with where $\|.\|$ denotes the Euclidean norm, K is a constant gain, and $\lambda > 0$ is a positive constant. (i.e., this non-affine and nonlinear dynamic system is exponentially stable).
- **Problem3:** (The problem of practical control) The core challenge lies in the fighter aircraft's ability to perform aggressive maneuvers that induce high angles of attack, necessitating a control strategy that directly regulates path angles and trajectory while effectively handling under-actuation. The mathematical model can thus be reformulated as: $d\tilde{\xi}/dt = \mathrm{f}\big(\tilde{\xi}\big) + \mathrm{g}\big(\tilde{\xi}\big)\mathrm{u} + \mathrm{d}$ where $\mathrm{u} = [\delta_A\ \delta_R\ \delta_E]^T \in \Re^3$, $\tilde{\xi} = [p\ q\ r\ \alpha\ \beta\ \phi\ \theta\ V]^T \in \Re^8$ and d is any disturbances/noises. In this case, the system model is reformulated as:

$$\left\{\begin{array}{l} \dot{\omega}(t) = \mathrm{f}_\omega(\xi) + A_\omega(\xi)\mathrm{u}(t) + \Delta_\omega(t) \\ \dot{\eta}(t) = \mathrm{f}_\eta(\xi) + A_0(\xi)\omega(t) + A_\eta(\xi)\mathrm{u}(t) + \Delta_\eta(t) \\ \dot{V}(t) = \mathrm{f_v}(\xi, \mathrm{u}) + A_\mathrm{v}(\xi)\delta_{TH} + +\Delta_\mathrm{v}(t) \end{array}\right. \tag{6}$$

with $\mathrm{u}(t)$ is the surface deflection = aerodynamic-inputs, $\delta_{TH}(t)$ is the throttle deflection = thrust input, $\delta(t) = [\mathrm{u}^T(t)\ \ \delta_{TH}(t)]^T \in \Re^4$ is the input vector, ω, η, V are the state vectors of the system and $\xi(t) = [\omega^T\ \ \eta^T\ \ V]^T = [p\ q\ r\ \alpha\ \beta\ \phi\ V]^T \in \Re^7$. The proposed control structure consists of hierarchically different interconnected subsystems.

$$\left\{\begin{array}{ll} \mathrm{u}(k) = \mathcal{C}_1\big(\omega(k), \omega_\mathrm{c}(k), \Delta_\omega(k)\big) & \text{fast control} \\ \delta_{TH}(k) = \mathcal{C}_3\big(\mathrm{x}(k), \mathrm{V_d}(k), \mathrm{u}(k-1), \Delta_V(k)\big) & \text{speed control} \\ \omega_\mathrm{c}(k) = \mathcal{C}_2\left(\eta(k), \eta_\mathrm{d}(k), \Delta_\eta(k)\right) & \text{slow control} \\ \eta_\mathrm{d}(k) = \mathcal{C}_\mathrm{GP}\big(\mathrm{r}(k), \mathrm{r_d}(k), \mathrm{V_d}(k)\big) & \text{G processor} \\ \Delta_\omega(k) = \mathrm{NN}(\mathrm{x}, \omega_\mathrm{c}); \quad \Delta_V(k) = \mathrm{NN}(\mathrm{x}, V_\mathrm{d}); \quad \Delta_\eta(k) = \mathrm{NN}(\mathrm{x}, \eta_d) & \end{array}\right\} \tag{7}$$

- **Goal 3:** The objective is to design a three-stage guidance/control law for the Su-30: an inner loop regulating velocity $[p\ \ q\ \ r]$, a middle loop managing $\eta(t) = [\alpha\ \beta\ \phi]^T$, and an outer loop generating the guidance law, which produces commanded $\eta_d(t) = [\alpha_d\ \beta_d\ \phi_d]^T$ from the desired trajectory. The strategy is validated through Software-in-the-Loop simulations using Microsoft Flight Simulator (FS2004). Table 3 summarizes the problem statements for the three control strategies, with their system dynamics and control architectures.

Table 3. Summary of problem formulations and control structures for the proposed control strategies

Symmetric model $\dot{\mathrm{x}} = \mathrm{f_{sym}}(\mathrm{x}, \delta) + \mathrm{d}$

$\begin{cases} \dot{\mathrm{x}}_1 = \mathrm{f_{sym1}}(\mathrm{x}_1, \mathrm{x}_2, \delta_{TH}) + \mathrm{g_{sym1}}(\mathrm{x}_1, \mathrm{x}_2)\mathrm{u} + \mathrm{d}_1 \\ \dot{\mathrm{x}}_2 = \mathrm{f_{sym2}}(\mathrm{x}_1, \mathrm{x}_2, \delta_{TH}) + \mathrm{d}_2 \end{cases}$

u = control surface deflection = aero_inputs
δ_{TH} = throttle deflection = thrust_input
$\delta = [\mathrm{u}^T\ \ \delta_{TH}]^T \in \Re^4$ = input vector
$\mathrm{x}_1, \mathrm{x}_2$ = state vectors of the system
$\mathrm{d}_1, \mathrm{d}_2$ = the disturbances vectors + noises
V, x, y, z = controlled states (speed/position)
$\mathrm{x}_1^T = [\mathrm{v}^T\ \ \omega^T] \in \Re^6$; $\mathrm{x}_2^T = [\mathrm{r}^T\ \ \vartheta^T] \in \Re^6$
$\mathrm{x}^T = [\mathrm{x}_1^T\ \ \mathrm{x}_2^T] = [u\ v\ w\ p\ q\ r\ \phi\ \theta\ \psi\ x\ y\ z]$

The symmetric model can be written as

$\begin{cases} \dot{V} = \mathrm{f}_V^{\mathrm{sym}}(\mathrm{x}) + \mathrm{f}_{1\mathrm{u}}^{\mathrm{sym}}(\mathrm{x})\mathrm{u} + \mathrm{f}_{1\mathrm{th}}^{\mathrm{sym}}(\mathrm{x})\delta_{TH} + \mathrm{d}_V \\ \dot{\omega} = \mathrm{f}_\omega^{\mathrm{sym}}(\mathrm{x}) + \mathrm{f}_{2\mathrm{u}}^{\mathrm{sym}}(\mathrm{x})\mathrm{u} + \mathrm{f}_{2\mathrm{th}}^{\mathrm{sym}}(\mathrm{x})\delta_{TH} + \mathrm{d}_\omega \\ \dot{\vartheta} = \mathrm{f}_\vartheta^{\mathrm{sym}}(\mathrm{x}) + +\mathrm{d}_\vartheta \end{cases}$

The proposed controller

$\mathrm{u}(t) = \mathcal{C}_1(\omega, \omega_\mathrm{c}, \delta_{TH}, \mathrm{d}_\omega)$	fast_mode controller
$\delta_{TH}(t) = \mathcal{C}_3(\mathrm{x}, \mathrm{V_d}, \mathrm{u}, \mathrm{d}_V)$	speed controller
$\omega_\mathrm{c}(t) = \mathcal{C}_2(\vartheta, \vartheta_\mathrm{d}, \mathrm{d}_\vartheta)$	slow_mode controller
$\vartheta_\mathrm{d}(t) = \mathcal{C}_\mathrm{GP}(\mathrm{r}, \mathrm{r_d}, \mathrm{V_d})$	guidance processor

$\mathrm{d}_V = \mathrm{NN}(\mathrm{x}, V_\mathrm{d})$; $\mathrm{d}_\omega = \mathrm{NN}(\mathrm{x}, \omega_\mathrm{c})$; $\mathrm{d}_\vartheta = \mathrm{NN}(\mathrm{x}, \vartheta_\mathrm{d})$

Asymmetric model $\dot{\mathrm{x}} = \mathrm{f_{asy}}(\mathrm{x}, \delta_{\mathrm{asy}}) + \mathrm{d}$

$\begin{cases} \dot{\mathrm{x}}_1 = \mathrm{f_{asy1}}(\mathrm{x}_1, \mathrm{x}_2) + \Pi(\mathrm{x}_1, \mathrm{x}_2)\delta_{\mathrm{asy}} + \mathrm{d}_1 \\ \dot{\mathrm{x}}_2 = \mathrm{f_{asy2}}(\mathrm{x}_1, \mathrm{x}_2) + \mathrm{d}_2 \end{cases}$

δ_{asy}: differential control deflections
$\delta_{\mathrm{asy}} = [\delta_{LA}\ \ \delta_{RA}\ \ \delta_R\ \ \delta_{LE}\ \ \delta_{RE}\ \ \delta_{TH}]^T \in \Re^6$

$\mathrm{x}_1, \mathrm{x}_2$ = state vectors of the system
$\mathrm{d}_1, \mathrm{d}_2$ = the disturbances vectors + noises
x_2 = controlled states (position/orient ...)

The asymmetric model can be written as

$\begin{cases} \dot{\mathrm{x}}_1 = \mathrm{f_{asy1}}(\mathrm{x}_1, \mathrm{x}_2) + \Pi(\mathrm{x}_1, \mathrm{x}_2)\delta_{\mathrm{asy}} + \mathrm{d}_1 \\ \dot{\mathrm{x}}_2 = \mathrm{T}(\mathrm{x})\mathrm{x}_1 + +\mathrm{d}_2 \end{cases}$

The proposed asymmetric controller

$\delta_{\mathrm{asy}}(t) = \mathcal{C}_1(\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_{1c}, \mathrm{d}_1)$	inner loop
$\mathrm{x}_{1c}(t) = \mathcal{C}_2(\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_{2d}, \mathrm{d}_2)$	outer loop

$\mathrm{d}_1 = \mathrm{NN}(\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_{1c})$; $\mathrm{d}_2 = \mathrm{NN}(\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_{2d})$
x_{1c} is the commanded state vector
$\mathrm{x}_{2d} = [\mathrm{r}_d^T\ \ \vartheta_d^T]^T$: the desired state vector

Practical model $\dot{\tilde{\xi}} = \mathrm{f}(\tilde{\xi}, \delta) + \mathrm{d}$ with $\tilde{\xi} = [p\ q\ r\ \alpha\ \beta\ \phi\ \theta\ \ V]^T \in \Re^8$

The practical model can be written as

$\begin{cases} \dot{\omega} = \mathrm{f}_\omega(\xi) + A_\omega(\xi)\mathrm{u} + \Delta_\omega \\ \dot{\eta} = \mathrm{f}_\eta(\xi) + A_0(\xi)\omega + A_\eta(\xi)\mathrm{u} + \Delta_\eta \\ \dot{V} = \mathrm{f_v}(\xi, \mathrm{u}) + A_\mathrm{v}(\xi)\delta_{TH} + +\Delta_\mathrm{v} \end{cases}$

u = surface deflection = aero_inputs
δ_{TH} = throttle defl = thrust_input
$\delta = [\mathrm{u}^T\ \ \delta_{TH}]^T \in \Re^4$ = input vector
$\xi = [\omega^T \eta^T V]^T = [p\ q\ r\ \alpha\ \beta\ \phi\ V]^T \in \Re^7$
ω, η, V = state vectors of the system
ξ = overall state vectors of the system

The proposed practical controller

$\mathrm{u}(t) = \mathcal{C}_1(\omega, \omega_\mathrm{c}, \Delta_\omega)$	fast control
$\delta_{TH}(t) = \mathcal{C}_3(\mathrm{x}, \mathrm{V_d}, \mathrm{u}, \Delta_V)$	speed control
$\omega_\mathrm{c}(t) = \mathcal{C}_2(\eta, \eta_\mathrm{d}, \Delta_\eta)$	slow control
$\eta_\mathrm{d}(t) = \mathcal{C}_\mathrm{GP}(\mathrm{r}, \mathrm{r_d}, \mathrm{V_d})$	G processor

$\Delta_\omega = \mathrm{NN}(\mathrm{x}, \omega_\mathrm{c})$;
$\Delta_V = \mathrm{NN}(\mathrm{x}, V_\mathrm{d})$;
$\Delta_\eta = \mathrm{NN}(\mathrm{x}, \eta_d)$;

3. New Symmetric Surface Control

Based on the time-scale separation and nonlinear control theory, the proposed guidance and control strategy is structured into guidance, attitude, and angular rate loops. Before detailing this approach, we first revisit the nonlinear dynamic inversion method and RBF neural networks and we ref the reader to see [7], [11], [25] and [28], [39].

3.1. Basics of Nonlinear Dynamic Inversion and RBF Neural Networks

Consider a MIMO nonlinear dynamical system in the form of $\dot{x} = \mathrm{f_N}(x) + \mathrm{g_N}(x)u$ and $z = \mathrm{h_N}(x)$ where $x \in \Re^n$, $u \in \Re^m$, $z \in \Re^m$, and $\mathrm{f_N}$, $\mathrm{g_N}$ and $\mathrm{h_N}$ are smooth functions of x. The problem consists of finding $u = \alpha(x) + \beta(x)\mathrm{v}$ which transform the system into an equivalent controllable linear system $\dot{x} = \mathcal{A}x + \mathcal{B}u$ from which the auxiliary control synthesis is performed [8], [3]. In simple case (assume the relative degree is one) we take the first derivative of the output $z(t)$ to obtain.

$$\dot{z}(t) = \frac{\partial \mathrm{h_N}}{\partial x}\left(\mathrm{f_N}(x) + \mathrm{g_N}(x)u(t)\right) = \frac{\partial \mathrm{h_N}}{\partial x}\mathrm{f_N}\left(x(t)\right) + \left\{\frac{\partial \mathrm{h_N}}{\partial x}\mathrm{g_N}(x)\right\}u(t) \tag{8}$$

The nonlinear dynamic inversion of the equation (8) is.

$$u = \left[\frac{\partial \mathrm{h_N}}{\partial x}\mathrm{g_N}(x)\right]^{-1}\left\{\dot{z}(t) - \frac{\partial \mathrm{h_N}}{\partial x}\mathrm{f_N}\right\} = \left[\frac{\partial \mathrm{h_N}}{\partial x}\mathrm{g_N}(x)\right]^{-1}\left\{\dot{z}_d(t) + \mathrm{K}e_z(t) - \frac{\partial \mathrm{h_N}}{\partial x}\mathrm{f_N}\right\} \tag{9}$$

The output $z(t)$ is controlled by linear first order system $\dot{z}(t) = \dot{z}_d(t) + \mathrm{K}e_z(t)$ so if we put $\mathrm{v} = z_d(t)$ we get $\beta(x) = [(\partial \mathrm{h_N}/\partial x)\mathrm{g_N}]^{-1}$, $\alpha(x) = [(\partial \mathrm{h_N}/\partial x)\mathrm{g_N}]^{-1}[(\partial \mathrm{h_N}/\partial x)\mathrm{f_N}(x)]$. A functional schematic diagram of this result is shown in Fig. 3.

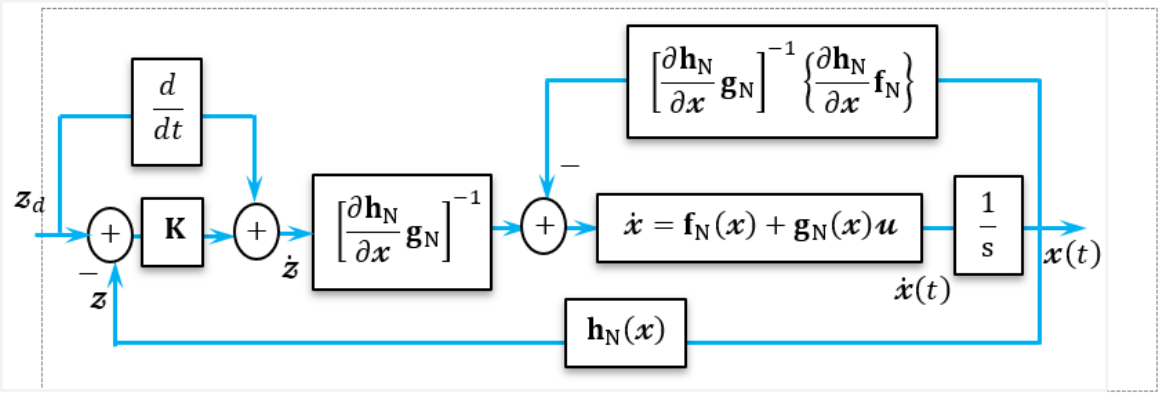

Fig. 3. Simple block diagram of nonlinear dynamic inversion

From the derivation above, it will be clear that Nonlinear Dynamic Inversion has a number of advantageous features [26], [35], [37]:

1. The method is straight-forward and easy to understand.
2. The system is fully decoupled, and the desired dynamics can be easily imposed.
3. Excellent nominal performance for closed-loop system.
4. Linear control is applied, since NDI cancels the nonlinear dynamics.

To approximate the unknown nonlinearities and un-modeled dynamics in the aircraft system, a Radial Basis Function Neural Network (RBFNN) is employed due to its universal approximation capability and fast convergence. RBFNNs are a specific class of feedforward neural networks that employ radial basis functions (typically Gaussian) as activation functions in their hidden layer [44]. They are grounded in the universal approximation theorem, which asserts that a sufficiently large

RBFNN can approximate any continuous nonlinear function over a compact domain to arbitrary precision. They are particularly favored in adaptive control due to their localized learning properties and fast convergence, while still satisfying global function approximation properties—i.e., they can represent highly nonlinear systems over the entire domain of operation. The RBFNN maps the input vector $x \in \mathbb{R}^n$ through $\phi_i(x) = \exp(-\|x - c_i\|^2/\sigma_i^2)$ a set of Gaussian basis functions, with c_i and σ_i are the center and width of the i^{th} neuron, respectively. The network output is given by $\hat{\mathrm{n}}(x) = W^T\phi(x)$, where $\phi(x) = [\phi_1(x), \ldots, \phi_N(x)]^T \in \mathbb{R}^N$ and $W \in \mathbb{R}^m$ is the weight vector. An adaptive law, typically gradient_based or Lyapunov-stable, updates the weights online to minimize the approximation error in real-time [7], [25]. This structure (Fig. 4) enables the controller to compensate for disturbances and uncertainties effectively, making it well-suited for aggressive maneuvering scenarios of Su-30 aircraft. The adaptive weight update law is derived to ensure uniform ultimate boundedness of tracking errors via Lyapunov stability theory. Let the RBFNN approximation be $\hat{\mathrm{n}}(\mathbb{x}) = W^{\star T}\phi(x)$, with the ideal weights $W^\star$, and define the weight estimation error as $\widetilde{W} = W - W^\star$. The adaptation law is: $\dot{W}(t) = \Gamma\phi(x)\mathrm{e}^T$ where $\Gamma > 0$ is the adaptation gain matrix, e is the tracking error [16].

3.2. Control of Conventional Surface Configuration

The standard six degrees of freedom nonlinear diff-equations for a well-known fixed-wing airplane can be summarized in the state space vector form as follows (see [40]-[42]).

$$\left\{\begin{matrix}\text{Forces and Moments} \\ \frac{d}{dt}\begin{bmatrix}\mathrm{v}\\ \omega\end{bmatrix} = \begin{bmatrix}(\mathrm{F_P}+\mathrm{F_A}+\mathrm{F_G})/m - \omega\times\mathrm{v}\\ \mathbb{I}_G^{-1}[\mathrm{M_P}+\mathrm{M_A}-\omega\times(\mathbb{I}_G.\omega)]\end{bmatrix}; \text{and}\end{matrix}\begin{matrix}\text{Trajectory and Attitude} \\ \frac{d}{dt}\begin{bmatrix}\mathrm{r}\\ \vartheta\end{bmatrix} = \begin{bmatrix}\mathrm{T_1}(\vartheta) & 0\\ 0 & \mathrm{T_0}(\vartheta)\end{bmatrix}\begin{bmatrix}\mathrm{v}\\ \omega\end{bmatrix}\end{matrix}\right\} \tag{10}$$

where the propulsive forces are $\mathrm{F_P} = (\delta_{th}\mathrm{F_{Pmax}})\mathrm{u_{Thrust}} \in \Re^{3\times1}$, the gravitational forces are $\mathrm{F_G} = m\mathrm{g_B} = m\mathrm{T_1}(\vartheta)\mathrm{g_o} \in \Re^{3\times1}$, the propulsive and aerodynamic forces (moments) are given by.

$$\mathrm{F_A} = Q\left\{\begin{bmatrix}\mathrm{C}_{x1}\\ \mathrm{C}_{y1}\\ \mathrm{C}_{z1}\end{bmatrix} + \begin{bmatrix}\mathrm{C}_{xa} & \mathrm{C}_{xr} & \mathrm{C}_{xe}\\ \mathrm{C}_{ya} & \mathrm{C}_{yr} & \mathrm{C}_{ye}\\ \mathrm{C}_{za} & \mathrm{C}_{zr} & \mathrm{C}_{ze}\end{bmatrix}\begin{bmatrix}\delta_a\\ \delta_r\\ \delta_e\end{bmatrix} + \begin{bmatrix}\mathrm{C}_{xT}\\ \mathrm{C}_{yT}\\ \mathrm{C}_{zT}\end{bmatrix}\delta_{th}\right\}; \qquad \mathrm{F_P} = \mathrm{F_T}\begin{bmatrix}\cos(\alpha_m)\cos(\beta_m)\\ \sin(\beta_m)\\ \sin(\alpha_m)\cos(\beta_m)\end{bmatrix}\delta_{th}$$

$$\mathrm{M_A} = Qs\left\{\begin{bmatrix}b\mathrm{C}_{l1}\\ c\mathrm{C}_{m1}\\ b\mathrm{C}_{n1}\end{bmatrix} + \begin{bmatrix}b\mathrm{C}_{la} & b\mathrm{C}_{lr} & b\mathrm{C}_{le}\\ c\mathrm{C}_{ma} & c\mathrm{C}_{mr} & c\mathrm{C}_{me}\\ b\mathrm{C}_{na} & b\mathrm{C}_{nr} & b\mathrm{C}_{ne}\end{bmatrix}\begin{bmatrix}\delta_a\\ \delta_r\\ \delta_e\end{bmatrix} + \begin{bmatrix}\mathrm{C}_{lT}\\ \mathrm{C}_{mT}\\ \mathrm{C}_{nT}\end{bmatrix}\delta_{th}\right\}; \qquad \mathrm{M_P} = \begin{bmatrix}x_\mathrm{m}\\ y_\mathrm{m}\\ z_\mathrm{m}\end{bmatrix}\times\mathrm{F_P}$$

with $Q = 0.5\rho V^2$, and α_m, β_m are the pitch and yaw setting, respectively and.

$$\mathrm{T_0}(\vartheta) = \begin{bmatrix}1 & \mathrm{s}_\phi\mathrm{s}_\theta/\mathrm{c}_\theta & \mathrm{c}_\phi\mathrm{s}_\theta/\mathrm{c}_\theta\\ 0 & \mathrm{c}_\phi & -\mathrm{s}_\phi\\ 0 & \mathrm{s}_\phi/\mathrm{c}_\theta & \mathrm{c}_\phi/\mathrm{c}_\theta\end{bmatrix}; \quad \mathrm{T_1}(\vartheta) = \begin{bmatrix}\mathrm{c}_\theta\,\mathrm{c}_\psi & \mathrm{c}_\theta\,\mathrm{s}_\psi & -\mathrm{s}_\theta\\ \mathrm{s}_\phi\,\mathrm{s}_\theta\,\mathrm{c}_\psi - \mathrm{c}_\phi\,\mathrm{s}_\psi & \mathrm{s}_\phi\,\mathrm{s}_\theta\,\mathrm{s}_\psi + \mathrm{c}_\phi\,\mathrm{c}_\psi & \mathrm{s}_\phi\,\mathrm{c}_\theta\\ \mathrm{c}_\phi\,\mathrm{s}_\theta\,\mathrm{c}_\psi + \mathrm{s}_\phi\,\mathrm{s}_\psi & \mathrm{c}_\phi\,\mathrm{s}_\theta\,\mathrm{s}_\psi - \mathrm{s}_\phi\mathrm{c}_\psi & \mathrm{c}_\phi\,\mathrm{c}_\theta\end{bmatrix}$$

$$\mathrm{v} = \begin{bmatrix}u\\ v\\ w\end{bmatrix}; \qquad \omega = \begin{bmatrix}p\\ q\\ r\end{bmatrix}; \qquad \mathrm{r} = \begin{bmatrix}x\\ y\\ z\end{bmatrix}; \qquad \vartheta = \begin{bmatrix}\phi\\ \theta\\ \psi\end{bmatrix}; \qquad \text{and} \qquad \mathbb{I}_G = \begin{bmatrix}I_{xx} & 0 & -I_{xz}\\ 0 & I_{yy} & 0\\ -I_{xz} & 0 & I_{zz}\end{bmatrix}$$

For simplicity, we use the notations $\mathrm{s}_\bullet$ and $\mathrm{c}_\bullet$ to denote, $\sin(\bullet)$ and $\cos(\bullet)$ respectively. In the case of conventional surface configuration (i.e. symmetric control) the dynamics of the Euler angles $\vartheta = [\phi \;\; \theta \;\; \psi]^T \in \Re^{3\times1}$ and inertial positions $\mathrm{r} = [x_B \;\; y_B \;\; z_B]^T \in \Re^{3\times1}$ are indirectly affected by the actuators $\delta(t) = [\delta_a \;\; \delta_r \;\; \delta_e \;\; \delta_{th}]^T \in \Re^{4\times1}$ through their tight coupling with velocities $\omega \in \Re^{3\times1}$ and $\mathrm{v} \in \Re^{3\times1}$. Now, it is well known that the four input variables $\delta(t)$ can only control four state variables. Therefore, we need to find judicious ways to control the essential outputs (i.e. position r and speed $V = \|\mathrm{v}\|$). The control inputs are three for surface deflections, δ_e (elevator), δ_a (aileron), δ_r (rudder), and the thrust throttle δ_{th}.

This section targets the design of an autopilot capable of tracking desired attitude, (consequently a desired trajectory), and velocity despite model uncertainties and parameter

variations. The aircraft's control structure (Fig. 5) suggests grouping control inputs as: $\mathrm{u}(t) = [\delta_a \ \delta_r \ \delta_e]^T \in \Re^{3\times1}$ and we make the following decomposition of the aero-propulsive forces are, $\mathrm{F_{AP}} = \mathrm{F_A} + \mathrm{F_P} = \mathrm{F_{A1}} + \Pi_{\mathrm{F1}}\delta_{th} + \Pi_{\mathrm{F2}}\mathrm{u}$ where.

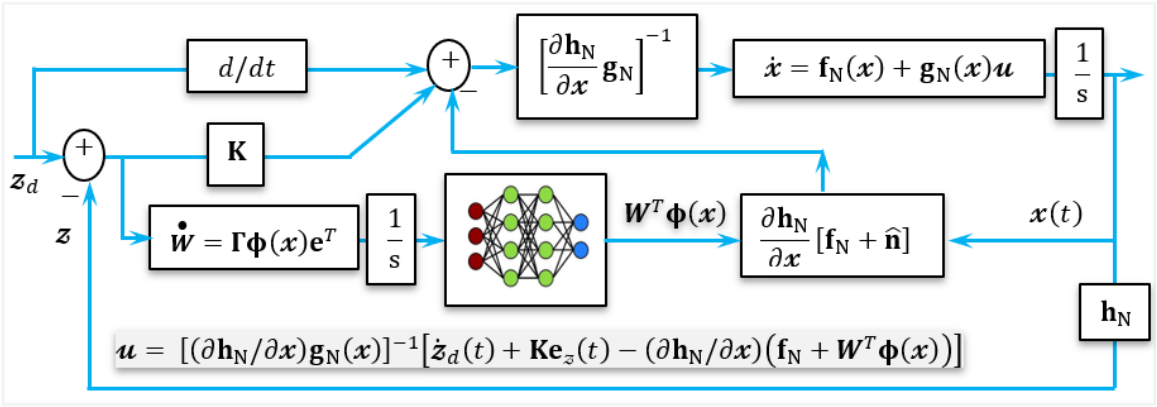

Fig. 4. Adaptive RBF neural control for NDI control system

$$\Pi_{\mathrm{F1}} = Qs\begin{bmatrix}\mathrm{C}_{xT}\\ \mathrm{C}_{yT}\\ \mathrm{C}_{zT}\end{bmatrix} + \mathrm{F_T}\begin{bmatrix}\cos(\alpha_m)\cos(\beta_m)\\ \sin(\beta_m)\\ \sin(\alpha_m)\cos(\beta_m)\end{bmatrix};\ \Pi_{\mathrm{F2}} = Qs\begin{bmatrix}\mathrm{C}_{xa} & \mathrm{C}_{xr} & \mathrm{C}_{xe}\\ \mathrm{C}_{ya} & \mathrm{C}_{yr} & \mathrm{C}_{ye}\\ \mathrm{C}_{za} & \mathrm{C}_{zr} & \mathrm{C}_{ze}\end{bmatrix};\ \mathrm{F_{A1}} = Qs\begin{bmatrix}\mathrm{C}_{x1}\\ \mathrm{C}_{y1}\\ \mathrm{C}_{z1}\end{bmatrix}$$

Likewise for the aero-propulsive moments $\mathrm{M_{AP}} = \mathrm{M_A} + \mathrm{M_P} = \mathrm{M_{A1}} + \Pi_{\mathrm{M1}}\delta_{th} + \Pi_{\mathrm{M2}}\mathrm{u}$.

$$\Pi_{\mathrm{M1}} = Qs\begin{bmatrix}\mathrm{C}_{lT}\\ \mathrm{C}_{mT}\\ \mathrm{C}_{nT}\end{bmatrix} + \mathrm{F_t}\begin{bmatrix}x_{\mathrm{m}}\\ y_{\mathrm{m}}\\ z_{\mathrm{m}}\end{bmatrix} \times \begin{bmatrix}\mathrm{c}_{\alpha_m}\mathrm{c}_{\beta_m}\\ \mathrm{s}_{\beta_m}\\ \mathrm{s}_{\alpha_m}\mathrm{c}_{\beta_m}\end{bmatrix};\ \Pi_{\mathrm{M2}} = Qs\begin{bmatrix}b\mathrm{C}_{la} & b\mathrm{C}_{lr} & b\mathrm{C}_{le}\\ c\mathrm{C}_{ma} & c\mathrm{C}_{mr} & c\mathrm{C}_{me}\\ b\mathrm{C}_{na} & b\mathrm{C}_{nr} & b\mathrm{C}_{ne}\end{bmatrix};\ \mathrm{M_{A1}} = Qs\begin{bmatrix}b\mathrm{C}_{l1}\\ c\mathrm{C}_{m1}\\ b\mathrm{C}_{n1}\end{bmatrix}$$

The dynamic model can be written as.

$$\frac{d}{dt}\begin{bmatrix}\mathrm{v}\\ \omega\end{bmatrix} = \begin{bmatrix}(\mathrm{F_{A1}} + \Pi_{\mathrm{F1}}\delta_{th} + \mathrm{F_G})/m \ - \omega\times\mathrm{v}\\ \mathbb{I}_G^{-1}\left(\mathrm{M_{A1}} + \Pi_{\mathrm{M1}}\delta_{th} - \omega\times(\mathbb{I}_G.\omega)\right)\end{bmatrix} + \begin{bmatrix}\Pi_{\mathrm{F2}}/m\\ \mathbb{I}_G^{-1}\Pi_{\mathrm{M2}}\end{bmatrix}\mathrm{u};\quad \frac{d}{dt}\begin{bmatrix}\mathrm{r}\\ \vartheta\end{bmatrix} = \begin{bmatrix}\mathrm{T}_1 & 0\\ 0 & \mathrm{T}_0\end{bmatrix}\begin{bmatrix}\mathrm{v}\\ \omega\end{bmatrix} \quad (11)$$

Or more compactly, the aircraft model is given by.

$$\dot{\mathrm{x}}_1 = \mathrm{f_{sym1}}(\mathrm{x}_1, \mathrm{x}_2, \delta_{th}) + \mathrm{g_{sym1}}(\mathrm{x}_1, \mathrm{x}_2)\mathrm{u} \quad \text{and} \quad \dot{\mathrm{x}}_2 = \mathrm{f_{sym2}}(\mathrm{x}_1, \mathrm{x}_2, \delta_{th}) \quad (12)$$

where $\mathrm{x}_1^T = [\mathrm{v}^T \ \ \omega^T] \in \Re^6$ and $\mathrm{x}_2^T = [\mathrm{r}^T \ \vartheta^T] \in \Re^6$ and.

$$\mathrm{f_{sym1}} = \begin{bmatrix}(\mathrm{F_{A1}} + \Pi_{\mathrm{F1}}\delta_{th} + \mathrm{F_G})/m - \omega\times\mathrm{v}\\ \mathbb{I}_G^{-1}\left(\mathrm{M_{A1}} + \Pi_{\mathrm{M1}}\delta_{th} - \omega\times(\mathbb{I}_G.\omega)\right)\end{bmatrix};\quad \mathrm{f_{sym2}} = \begin{bmatrix}\mathrm{T}_1 & 0\\ 0 & \mathrm{T}_0\end{bmatrix}\begin{bmatrix}\mathrm{v}\\ \omega\end{bmatrix};\quad \mathrm{g_{sym1}} = \begin{bmatrix}\Pi_{\mathrm{F2}}/m\\ \mathbb{I}_G^{-1}\Pi_{\mathrm{M2}}\end{bmatrix}$$

Military platforms (i.e. missiles, including fighter jets, drones, helicopters, and multi-rotors) use various engine types: piston, rotary, turboshaft, turbojet, or turboprop. Engine selection depends on aircraft weight, size, range, payload, and stealth. In this work, we consider a jet-propulsion model producing thrust for lift, cruise, and maneuvering [43]. The thrust force is given by: $\mathrm{F_{prop}} = \dot{m}(\mathrm{v}_e - \mathrm{v}_0) + A_e(P_e - P_0)$ where $\mathrm{F_{prop}}$ is the thrust force, A_e is the exhaust area, $\dot{m}$ is mass flow rate, v_e and v_0 are exhaust and inlet velocities, and P_e, P_0 are exhaust and ambient pressures, respectively. A simplified engine dynamics model is: $\tau[d\mathrm{F_{prop}}/dt] + \mathrm{F_{prop}} = K\delta_{th}(t)$ where: $\delta_{th}(t)$ is the throttle input, τ is the time constant, K is the steady-state gain relating throttle input to thrust output.

In applications in which measured atmospheric data are available at only one or a few altitudes h, the atmosphere is modeled in a flight simulation by using equations that extrapolate or interpolate data according to known principles of atmospheric variation with h [41]. These equations are.

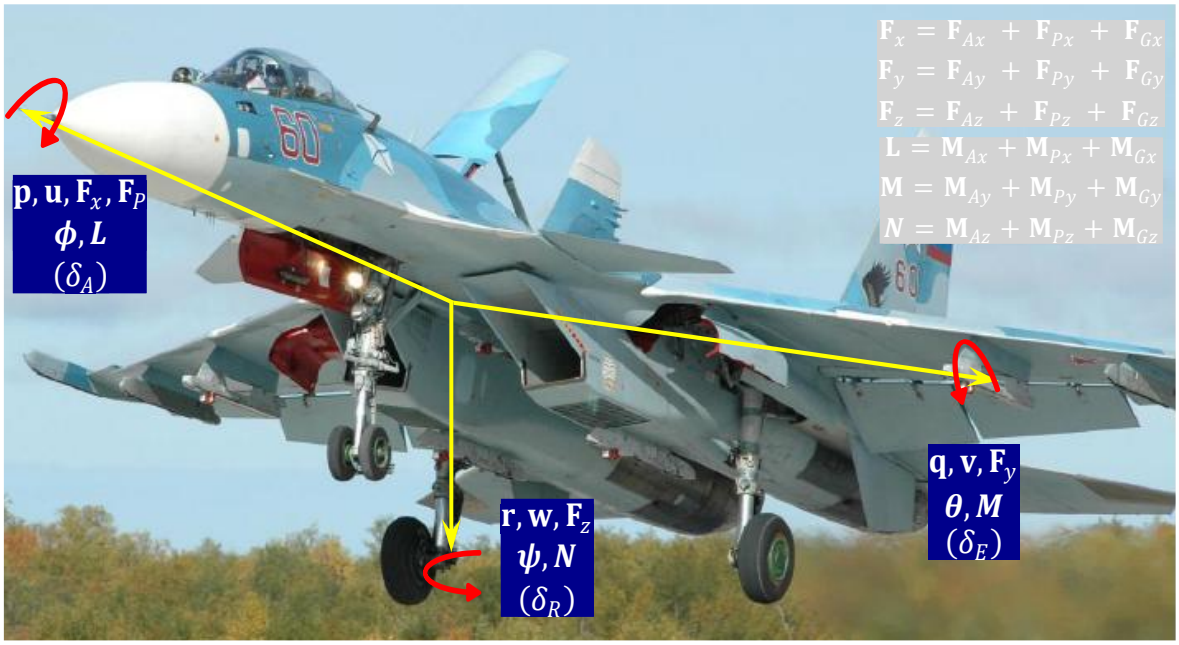

Fig. 5. Side view of a Sukhoi "Su-30MKA" aircraft in space

$$T = T_1 + a(h - h_1), [\mathrm{K}]. \quad P = P_1 + (T/T_1)^{-g/(aR)}, [\mathrm{Pa}]$$
$$\rho = P/(RT), [\mathrm{kg/m^3}]. \quad V_s = \sqrt{\gamma RT}, [\mathrm{m/s}] \tag{13}$$

Where a: lapse rate, [K/m]. ρ: atmospheric density, [kg/m^3]. g: magnitude of gravity at earth surface $\vec{g}$, [m/s^2]. h: altitude above sea level, [m], h_1: reference altitude, [m]. P_1: given pressure at altitude h_1, [Pa] and P: is at altitude h. R: gas constant (287.05), [N.m]/(kg.K). T: temperature at altitude h, [K] and T_1: given temperature at altitude h_1, [K]. V_s: speed of sound at altitude h, [m/s] and γ: ratio of specific heat, [dimensionless]. Some authors use the model (i.e. approximate version) $P = P_1 \exp[-g(h - h_1)/RT]$. And now, we give a detailed representation of the components and subsystems included in the aircraft model (see Fig. 6).

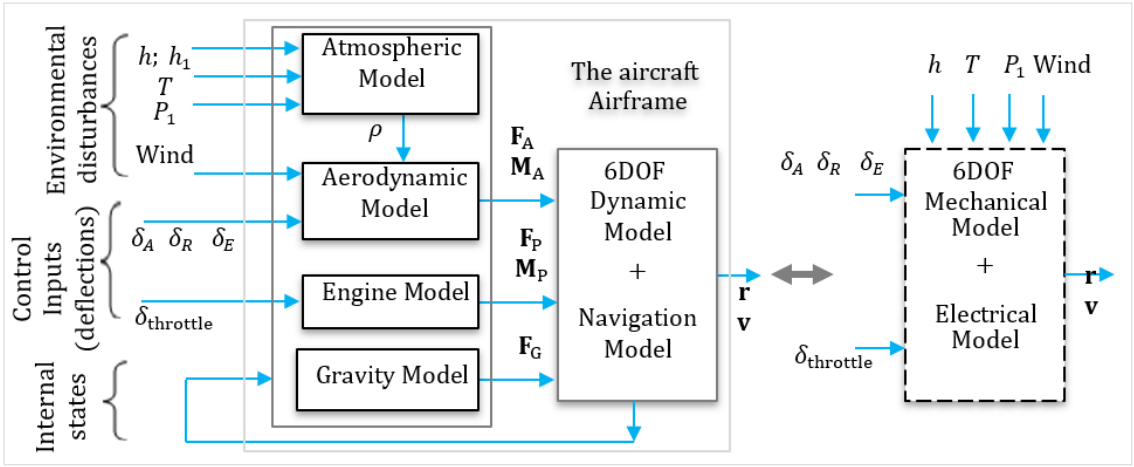

Fig. 6. Diagram of the subsystems included in Su-30MK model

3.2.1. Auto-Piloting and Controller Design

The first problem to be addressed is determination of the required angular rate commands to allow the vehicle to follow a desired path, we next formulate the inner-loop control problem with the goal of finding a system input which allows the vehicle to track the angular-rate commands given by the outer-loop controller (Fig. 7).

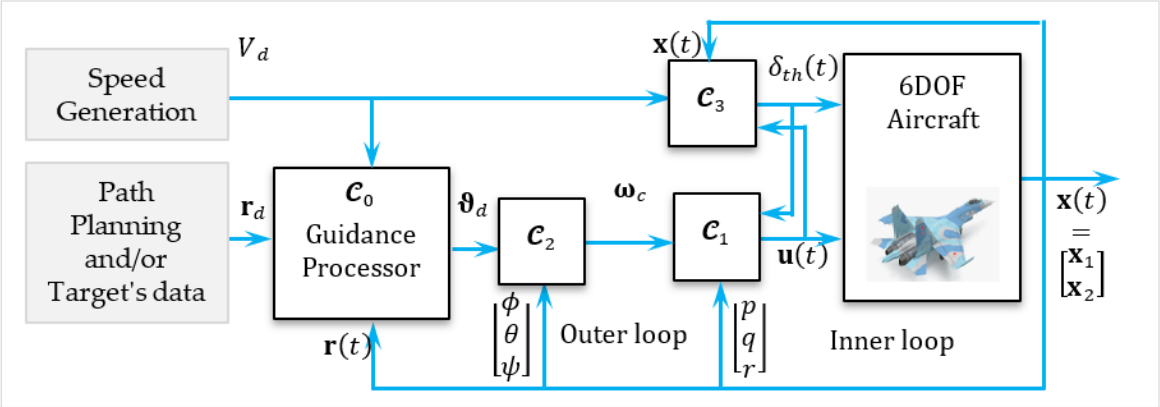

Fig. 7. Structural schematic diagram of the non-adaptive symmetric auto-pilot-controller

In Fig. 7, the inner-loop governs the fast dynamics (p, q, r), controlled by the rudder, elevators, and ailerons. The outer-loop handles the slower attitude dynamics (ϕ, θ, ψ), driven by the body rates (p, q, r). Since fast dynamics settle rapidly, their transients are assumed negligible in the outer-loop design. The control strategy proceeds as follows:

- Outer-loop ($\mathcal{C}_2$): Generates virtual rate ω_c to track desired ϑ_d, from a ref-path r_d.
- Inner-loop ($\mathcal{C}_1$): Drives control input u to make $\omega \to \omega_c$.
- Speed control ($\mathcal{C}_3$): Use the throttle input δ_{th} to follow a target speed profile V_d.
- Guidance processor ($\mathcal{C}_0$): Extracts ϑ_d from path r_d via virtual target-following.

Each controller is developed and detailed in the following subsections.

1) Two Loops Attitude Control: Now the above model is suitable for use in NDI control for the purpose of attitude and speed autopilot design. Assume that the output need to be controlled is the direction $\mathrm{y} = \vartheta(t)$ means that $\dot{\vartheta}(t) = \mathrm{T}_0(\vartheta)\omega$ or in term of the slow-dynamics inversion formula $\omega(t) = \mathrm{T}_0^{-1}\dot{\vartheta}(t)$. Now if we impose the control $\dot{\vartheta}(t) = \dot{\vartheta}_d(t) + \mathrm{K}_2\mathrm{e}_\vartheta(t)$ with an error vector $\mathrm{e}_\vartheta(t) = \vartheta_d - \vartheta$ then the outer-loop response is given by $\omega_c(t) = \mathrm{T}_0^{-1}\left[\dot{\vartheta}_d(t) + \mathrm{K}_2\mathrm{e}_\vartheta(t)\right]$. Let $\mathrm{K}_2 = \mathrm{diag}\left(\left[\lambda_\phi\ \lambda_\theta\ \lambda_\psi\right]\right) \in \Re^{3\times3}$ then the bandwidths λ_ϕ, λ_θ and λ_ψ should be slightly below or equal to the bandwidths of the aircraft dynamics. Assume that the fighter aircraft is equipped with an inner-loop controller that provides tracking capabilities of feasible roll, pitch and yaw-rate commands. In other words, $\|\omega_c(t) - \omega(t)\| \le \epsilon_\omega,\ \forall t \ge 0$ if $\omega_c(t)$ satisfy $\|\omega_c(t)\| \le \omega_{cmax},\ \forall t \ge 0$. The problem is to derive control laws for $\omega_c(t)$, such that $\mathrm{e}_\omega(t) = \omega_c(t) - \omega(t)$ converges to a neighborhood of the origin. For the inner loop we need to get the inverse dynamics between $\omega(t)$ and $\mathrm{u}(t)$ which is the vector of surface deflections. The fast-state dynamics are modeled by the matrix-form equation $\dot{\omega}(t) = \mathbb{I}_G^{-1}\left(\mathrm{M}_{A1} + \Pi_{M1}\delta_{th} - \omega_c \times (\mathbb{I}_G.\omega_c) + \Pi_{M2}\mathrm{u}(t)\right)$. Based on the nonlinear NDI method we can derive the deflection control signals as.

$$\mathrm{u}(t) = [\Pi_{M2}]^{-1}\{\mathbb{I}_G.\dot{\omega}(t) - \mathrm{f}_\mathrm{u}(\omega_c, \delta_{th})\} \tag{14}$$

where $\mathrm{f}_\mathrm{u}(\omega_c, \delta_{th}) = \mathrm{M}_{A1} + \Pi_{M1}\delta_{th} - \omega_c \times (\mathbb{I}_G.\omega_c)$. Now as before, we can replace the velocity ω by its equivalent controlled angular velocity $\omega_c(t)$ with $\dot{\omega}(t) = \dot{\omega}_c(t) + \mathrm{K}_1\mathrm{e}_\omega(t)$ so we obtain.

$$\mathrm{u}(t) = [\Pi_{M2}]^{-1}[\mathbb{I}_G\{\dot{\omega}_c(t) + \mathrm{K}_1\mathrm{e}_\omega(t)\} - \mathrm{f}_\mathrm{u}(\omega_c, \delta_{th})] \tag{15}$$

Equation (15) represents the inverse system of the fast states dynamics, which yields the desired control surface commands. Let $\mathrm{K}_1 = \mathrm{diag}\left(\left[\lambda_p\ \ \lambda_q\ \ \lambda_r\right]\right)$ then band-widths λ_p, λ_q and λ_r should sufficiently exceed the bandwidths of the outer-loop ϕ, θ and ψ loop to avoid coupling between the inner- and outer-loop dynamics. The bandwidth corresponds to the body rate cut off

frequencies. In control systems for fighter aircrafts, a limiter is often used between the servo (or actuator) model and the aerodynamic model to ensure that the control surfaces remain within physically feasible and aerodynamically safe limits $\mathrm{u_{min}} \leq \mathrm{u}(t) \leq \mathrm{u_{max}}$ i.e. $\mathrm{u_{new}} = \mathrm{clamp}(\mathrm{u}(t), \mathrm{u_{min}}, \mathrm{u_{max}})$.

Note: For practical implementation, first-order actuator dynamics are modeled as $H(s) = \delta/\delta_c = K_a/[1 + \tau_a s]$ (see Fig. 8). These dynamics are omitted in the control design, assuming actuators are fast relative to the control bandwidth, i.e., $1/\tau_a$ exceeds the closed-loop bandwidth.

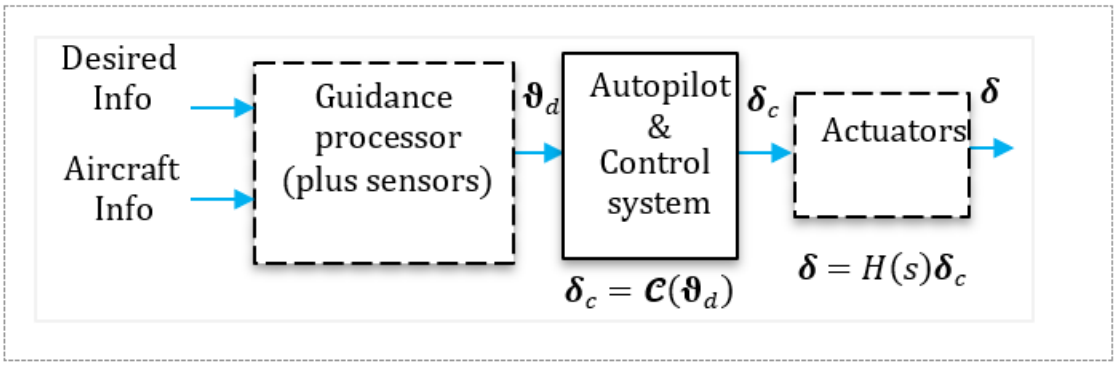

Fig. 8. Simple block diagram of the autopilot/control function

2) Speed Control: The NDI described above is used only for attitude control and none of them works for the velocity control. In considering chasing capabilities of an aircraft, one must take into account its velocity control, which is necessary to keep a desired distance from the target aircraft. For the aircraft speed control, we take into accounts some practical consideration of piloting. In effect, the pilot does not control the individual components of the velocity but rather its magnitude or norm: $V(t) = \|\mathrm{v}\| = [\mathrm{v}^T\mathrm{v}]^{1/2}$, so $\dot{V}(t) = \mathrm{v}^T\dot{\mathrm{v}}/V = \mathrm{v}^T(\mathrm{f}_{11} + \mathrm{g}_{11}\mathrm{u})/V$, $\mathrm{g}_{11} = \Pi_{\mathrm{F2}}/m$, $\mathrm{f}_{11} = (\mathrm{F_{A1}} + \Pi_{\mathrm{F1}}\delta_{th} + \mathrm{F_G})/m - (\omega \times \mathrm{v})$. Otherwise $\mathrm{v}^T(\omega \times \mathrm{v}) = 0$, and $\mathrm{u} = \mathrm{u}_c$ therefore.

$$\dot{V} = \frac{\mathrm{v}^T\dot{\mathrm{v}}}{V} = \left[\frac{\mathrm{v}^T}{mV}(\mathrm{F_{A1}} + \Pi_{\mathrm{F2}}\mathrm{u} + \mathrm{F_G}) + \left\{\frac{\mathrm{v}^T\Pi_{\mathrm{F1}}}{mV}\right\}\delta_{th}\right] \tag{16}$$

The inverse system of the speed dynamics, is.

$$\delta_{th} = \left[\frac{\mathrm{v}^T\Pi_{\mathrm{F1}}}{m\|\mathrm{v}\|}\right]^{-1}\left[\frac{d\|\mathrm{v}\|}{dt} - \frac{\mathrm{v}^T}{m\|\mathrm{v}\|}(\mathrm{F_{A1}} + \Pi_{\mathrm{F2}}\mathrm{u}_c + \mathrm{F_G})\right] \tag{17}$$

We impose that $\dot{V} = K_V e_v$ where $e_v = V_d - V$. Next, the pilot has to ensure that the aircraft does not stall. For this, we have to ensure that the derivative of r does not escape to infinity. As a result, we impose the condition: $\|\dot{\mathrm{r}}\|^2 \leq M$ where $M > 0$. Note that, $\mathrm{T}_1^{\mathrm{T}}(\vartheta) = \mathrm{T}_1^{-1}(\vartheta)$ is orthogonal, then $\|\dot{\mathrm{r}}\|^2 = \dot{\mathrm{r}}^T\dot{\mathrm{r}} = \mathrm{v}^T\mathrm{T}_1^{\mathrm{T}}(\vartheta)\mathrm{T}_1(\vartheta)\mathrm{v} = \mathrm{v}^T\mathrm{v} = V^2$. Therefore, the non-stalling condition reduces to ensure that $V \leq V_{\mathrm{max}}$, (i.e., we should protect the throttle by a limiter: $\delta_{\mathrm{min}} \leq \delta_{th} \leq \delta_{\mathrm{max}}$). Although high-performance engines respond quickly, typical low-cost engines have slower dynamics, hindering velocity and distance control. Thus, engine lag must be modeled, typically as a first-order $\mathrm{F_T}(s) = \mathrm{F_{Td}}(s)/(1 + \tau_E s)$. According to [28], the gain K_V is: $K_V = (1 - \tau_E X_u)^2/3\tau_E$ where τ_E is the engine time constant and X_u is the aerodynamic derivative in the wind-axis.

3) Guidance Law: Consider the following desired trajectory as $\mathrm{r_d} = [x_d \;\; y_d \;\; z_d]^T$ which is formulated in the inertial reference frame with the north-east-down orientation. In addition, it is assumed that the origin of the inertial coordinate system is coincident with the aircraft center of gravity (CG) at the beginning of the flight simulation. Furthermore, the displacement vector of the aircraft in the inertial reference frame is given by $\mathrm{r} = [x_I \;\; y_I \;\; z_I]^T$ and the velocity vector of the aircraft in such frame is $\mathrm{v} = [v_x \;\; v_y \;\; v_z]^T$. An effective and general guidance law for 3D trajectory

tracking, which is based on the *virtual target-following* approach, is used in this paper to determine the desired system output based on the desired trajectory components. In this approach, the guidance law is designed such that the vehicle tracks a moving target along the desired trajectory. More precisely, the instantaneous target point is defined as follows: $\mathrm{r}_T(t) = \mathrm{r}_\mathrm{d}(t+d)$ with.

$$\mathrm{r}_\mathrm{d}(t+d) = [x_d(t+d) \quad y_d(t+d) \quad z_d(t+d)]^T \tag{18}$$

The d is a positive constant parameter, which is used to adjust the convergence speed. Now, the desired pitch and yaw rates are determined in such a way that the velocity vector of the air vehicle rotates toward the relative position of the instantaneous target in each iteration. More precisely.

$$\begin{aligned}\dot{\theta}_d(t) &= -K_\theta\left[\sin^{-1}\left\{\frac{z_d(t+d)-z_I(t)}{\|\mathrm{r}_T(t)-\mathrm{r}(t)\|}\right\}+\sin^{-1}\left\{\frac{v_z(t)}{\|V(t)\|}\right\}\right]\\ \dot{\psi}_d(t) &= -K_\psi\left[\tan^{-1}\left\{\frac{y_d(t+d)-y_I(t)}{x_d(t+d)-x_I(t)}\right\}+\tan^{-1}\left\{\frac{v_y(t)}{v_x(t)}\right\}\right]\end{aligned} \tag{19}$$

where K_θ and K_ψ are some positive constant parameters. To determine the desired value of the roll rate, note that the instantaneous centrifugal force of the air vehicle should be generated through the horizontal component of the lift force. Thus, we have $\phi_d(t) = K_\phi \arctan\left(\dot{\psi}_d(t)v_{xy}/\mathrm{g}\right)$ where $v_{xy} = \sqrt{\left(v_x^2 + v_y^2\right)}$ and K_ϕ is a positive constant [30], [44]. So, we summarize the structure of the proposed controllers:

$$\begin{aligned}&\mathcal{C}_0\text{: Guidance processor provides } \vartheta_d \text{ from trajectory } \mathrm{r}_d\\ &\mathcal{C}_1\text{:}\quad \mathrm{u}(t) = [\Pi_{\mathrm{M2}}]^{-1}\left[\mathbb{I}_G\left\{\frac{d\omega_c}{dt}+\mathrm{K}_1\mathrm{e}_\omega(t)\right\}-\mathrm{f}_\mathrm{u}(\omega_c,\delta_{th})\right]\\ &\mathcal{C}_2\text{:}\quad \omega_c(t) = \mathrm{T}_0^{-1}\left[\dot{\vartheta}_d(t)+\mathrm{K}_2\mathrm{e}_\vartheta(t)\right]\\ &\mathcal{C}_3\text{:}\quad \delta_{th} = \left[\frac{\mathrm{v}^T\Pi_{\mathrm{F1}}}{m\|\mathrm{v}\|}\right]^{-1}\left[K_V e_v - \frac{\mathrm{v}^T}{m\|\mathrm{v}\|}(\mathrm{F_{A1}}+\Pi_{\mathrm{F2}}\mathrm{u}+\mathrm{F_G})\right]\end{aligned} \tag{20}$$

The outer loop corresponds to the slow-state dynamics ϕ, θ, ψ are controlled by p, q, r. Regarding the outer loop, the transient dynamics of the fast states occur so quickly that they have a negligible effect on the slow states. The fast loop dynamics p, q, r are controlled by $\delta_a(t)$, $\delta_e(t)$, $\delta_r(t)$. The desired closed-loop fast dynamics is $\dot{\omega}(t) = \dot{\omega}_c(t) + \mathrm{K}_1\mathrm{e}_\omega(t)$ where $\mathrm{K}_1 = \mathrm{diag}\left(\left[\lambda_p, \lambda_q, \lambda_r\right]\right)$ and λ_p, λ_q, λ_r are the bandwidth frequency set as high as they can be without exciting structural modes or being subject to the bandwidth limitations of the control actuators. The frequency of the roll, pitch and yaw channels is set to be $\lambda_p = \lambda_q = \lambda_r = 10$ [Hz] to ensure the avoiding of coupling between inner and outer loop. The inner loop simulation focuses on the ability of the NDI control laws to degrade the coupling between roll and yaw channels. In addition to that, the performance of tracking sever flight maneuvers is analyzed by drawing the $3D$ trajectory by initially starting with $x = y = h = 0$ [m]. The desired closed-loop slow dynamics are $\dot{\vartheta}(t) = \dot{\vartheta}_d(t) + \mathrm{K}_2\mathrm{e}_\vartheta(t)$ where $\mathrm{K}_2 = \mathrm{diag}\left(\left[\lambda_\phi,\ \lambda_\theta,\ \lambda_\psi\right]\right)$, $\lambda_\phi = 2;\ \lambda_\theta = 1;\ \lambda_\psi = 1.5$ [Hz] are the bandwidth frequency. These frequency gains are small enough compared with λ_p, λ_q, λ_r which guarantees to avoid coupling with inner fast loop. The outer attitude loop focuses on the outer loop tracking commands besides the inner fast loop response to high rate of change control commands.

4) Adaptation Mechanism via Neural Nets: Now, we present the theory behind the Feedback Error Learning (FEL) technique, which embeds neural networks into adaptive control, with application to flight control. Consider the non-affine model of an aircraft:

$$\left\{\begin{aligned}\frac{d\|\mathrm{v}\|}{dt} &= \frac{\mathrm{v}^T}{m\|\mathrm{v}\|}[\mathrm{F_{A1}}+\Pi_{\mathrm{F2}}\mathrm{u}+\Pi_{\mathrm{F1}}\delta_{th}+\mathrm{F_G}]+\Delta_{11}\\ \frac{d\omega}{dt} &= \mathbb{I}_G^{-1}[\mathrm{M_{A1}}+\Pi_{\mathrm{M1}}\delta_{th}-\omega\times(\mathbb{I}_G.\omega)+\Pi_{\mathrm{M2}}\mathrm{u}(t)]+\Delta_{12}\\ \frac{d\vartheta}{dt} &= \mathrm{T}_0\omega(t)+\Delta_{22}\end{aligned}\right\} \tag{21}$$

where ω, ϑ, $u \in \Re^3$, $\|v\| \in \Re$ and $\Delta_{11}(t)$, $\Delta_{12}(t)$ and $\Delta_{22}(t)$ stands for model uncertainties and external disturbances. Defining the desired references V_d, ω_c, and ϑ_d the tracking error is obtained as $e_\omega(t) = \omega_c(t) - \omega(t)$, and $e_\vartheta(t) = \vartheta_d(t) - \vartheta(t)$ and $e_v(t) = V_d(t) - V(t)$. Now, the control commands are.

$$\left\{\begin{array}{l} \delta_{th} = \left[\dfrac{v^T \Pi_{F1}}{m\|v\|}\right]^{-1}\left\{K_V e_v - \dfrac{v^T f_v(\vartheta, u)}{m\|v\|} - \Delta_{11}\right\} \\ u(t) = [\Pi_{M2}]^{-1}\left[\mathbb{I}_G\left\{\dfrac{d\omega_c}{dt} + K_1 e_\omega(t)\right\} - f_u(\omega_c, \delta_{th}) - \Delta_{12}\right] \\ \omega_c(t) = T_0^{-1}\left[\dfrac{d\vartheta_d}{dt} + K_2 e_\vartheta(t) - \Delta_{22}\right] \end{array}\right\} \quad (22)$$

with $f_v(\vartheta, u) = (F_{A1} + \Pi_{F2} u + F_G), f_u(\omega_c, \delta_{th}) = M_{A1} + \Pi_{M1}\delta_{th} - \omega_c \times (\mathbb{I}_G . \omega_c)$. However, the vector Δ_{12} and Δ_{22} are unknowns. Thus, they are approximated by a neural network NN (such as RBFNN or multilayer perceptron) as $\Delta = W^T \mu(x)$, where μ represents the vector of basis functions (corresponding to hidden layers of the NN) and W indicates the matrix of unknown weights which should be identified. Such a formulation can be used to represent different feedforward and recurrent NNs. Accordingly, due to the universal approximations property of neural networks, we have: $\hat{\Delta} = W^{\star T}\mu(x) + \varepsilon$ where $W^\star$ is the unknown optimal weight and ε indicates the bounded approximation error ($\|\varepsilon\| \leq \varepsilon_M$). Now, consider a Lyapunov function as $E_K = \left[e^T(t)e(t) + \text{trace}\left(E_W^T \Gamma^{-1} E_W\right)\right]/2$ where $E_W = W - W^\star$ and Γ is a positive definite matrix. Next, we have.

$$\frac{dE_K}{dt} = e^T(t)\frac{de}{dt} + \text{trace}\left(E_W^T \Gamma^{-1}\frac{dE_W}{dt}\right) = -e^T(t)\{e_\Delta + Ke\} + \text{trace}\left(E_W^T \Gamma^{-1}\frac{dE_W}{dt}\right) \quad (23)$$

where, $e_\Delta = \hat{\Delta} - \Delta$. Accordingly, defining $\dot{W} = \Gamma \mu e^T$ and considering $\dot{E}_W = \dot{W}$ (as a consequence of assuming a constant optimal weight $W^\star$, while such an assumption is reasonable even in the case of a time dependent uncertain term with $\dot{W}^\star \ll \dot{W}$), we have $dE_K/dt = -e^T(Ke - \varepsilon)$ which leads to $dE_K/dt < 0$ for $\|Ke\| > \|\varepsilon\|$ thereby guaranteeing the bounded tracking error. Selecting optimal design parameters such as K and Γ is typically challenging and often relies on trial and error. A more systematic approach formulates an optimization problem over these parameters and solves it using established methods (e.g., evolutionary algorithms) based on predefined criteria [7], [25], [39] and [42]. The adaptive intelligent control commands are then defined as:

$$\begin{array}{ll} \mathcal{C}_0: & \text{Guidance processor provides } \vartheta_d \text{ from trajectory } r_d \\ \mathcal{C}_1: & u(t) = [\Pi_{M2}]^{-1}\left[\mathbb{I}_G\left\{\dfrac{d\omega_c}{dt} + K_1 e_\omega(t)\right\} - f_u - W_\omega^T \mu_\omega\right] \\ \mathcal{C}_2: & \omega_c(t) = T_0^{-1}\left[\dot{\vartheta}_d(t) + K_2 e_\vartheta(t) - W_\vartheta^T \mu_\vartheta\right] \\ \mathcal{C}_3: & \delta_{th} = \left[\dfrac{v^T \Pi_{F1}}{m\|v\|}\right]^{-1}\left[K_V e_v - \dfrac{v^T}{m\|v\|}(f_v(\vartheta, u) - W_v^T \mu_v)\right] \end{array} \quad (24)$$

With

$f_u(\omega_c, \delta_{th}) = M_{A1} + \Pi_{M1}\delta_{th} - \omega_c \times (\mathbb{I}_G . \omega_c)$; $f_v(\vartheta, u) = (F_{A1} + \Pi_{F2} u + F_G)$;

$\dfrac{dW_\vartheta}{dt} = \Gamma_\vartheta \mu_\vartheta e_\vartheta^T$; $\dfrac{dW_\omega}{dt} = \Gamma_\omega \mu_\omega e_\omega^T$; and $\dfrac{dW_v}{dt} = \Gamma_v \mu_v e_v^T$

The overall view of the proposed adaptive symmetric solution is shown in Fig. 9.

3.2.2. Aerodynamic Coefficients Identification

Accurate modeling of aircraft behavior under various flight conditions relies on aerodynamic coefficients in F_A and M_A, which are often unknown or classified. To identify them effectively, this

study employs the Total Least Squares Estimation (TLSE) method, as detailed in [45], [46]. Based on Singular Value Decomposition (SVD), TLSE can robustly estimate matrix inverses, even when singular or non-square [47]. Aerodynamic data (or coefficients) are represented by force vector $\mathrm{F_A} = [F_{Ax};\ F_{Ay};\ F_{Az}]$ and moment vector $\mathrm{M_A} = [L;\ M;\ N]$ (roll, pitch, yaw).

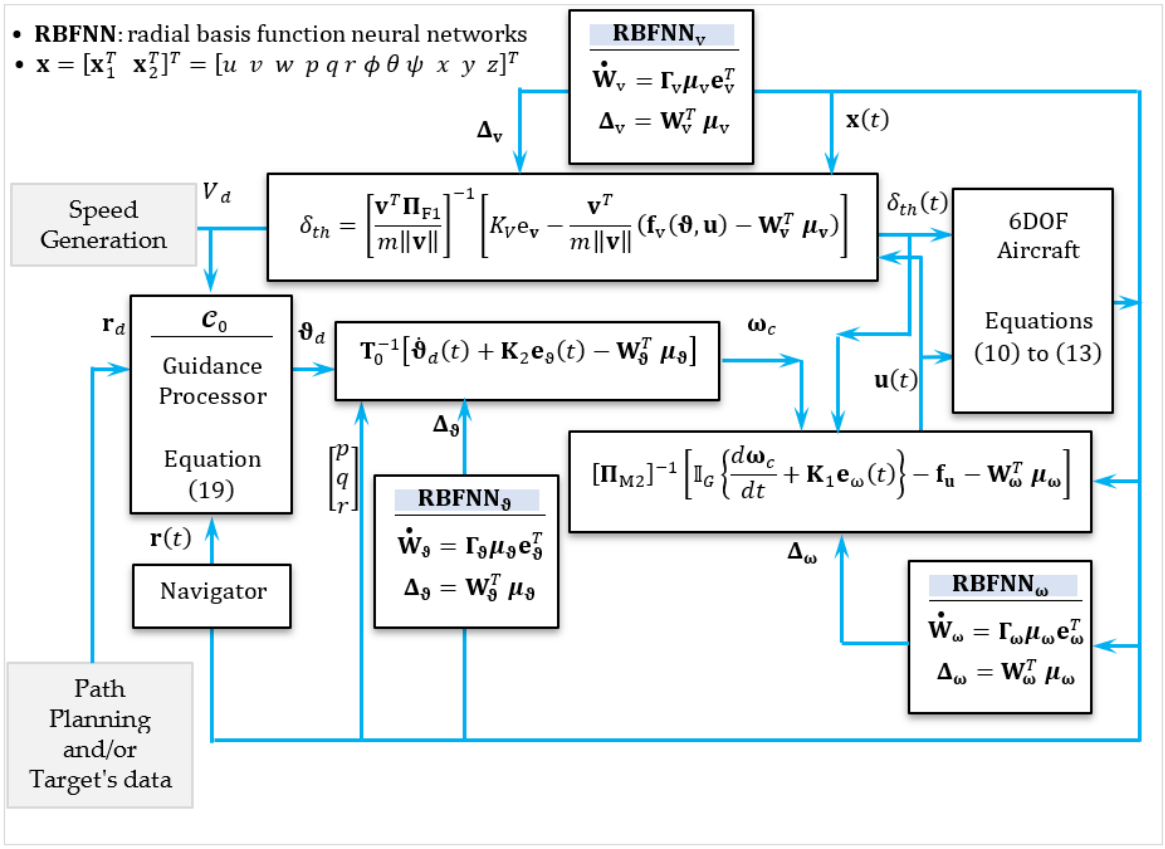

Fig. 9. Detailed structural form of the adaptive symmetric auto-pilot-controller function

$$C_x = \frac{F_{Ax}}{Qs};\quad C_y = \frac{F_{Ay}}{Qs};\quad C_z = \frac{F_{Az}}{Qs};\quad C_l = \frac{L/b}{Qs};\quad C_m = \frac{M/c}{Qs};\quad C_n = \frac{N/b}{Qs};\quad Q = 0.5\rho V^2 \tag{25}$$

where F_{Ax}, F_{Ay}, F_{Az}, L, M, and N, are the aerodynamic forces and moments obtained in the body reference frame [40], [41].

$$\begin{cases} F_{Ax} = m(\dot{u} - rv + qw) - (F_T c_{\alpha m} c_{\beta m} - \mathrm{g}s_\theta) \\ F_{Ay} = m(\dot{v} - pw + ru) - (F_T c_{\alpha m} s_{\beta m} + \mathrm{g}s_\phi c_\theta) \\ F_{Az} = m(\dot{w} - qu + pv) - (F_T s_{\alpha m} + \mathrm{g}c_\phi c_\theta) \end{cases} \quad \begin{cases} L = \dot{p}I_x - \dot{r}I_{xz} + qr(I_{zz} - I_{yy}) - pqI_{xz} - M_{Tx} \\ M = \dot{q}I_y + pr(I_{xx} - I_{zz}) + (p^2 - r^2)I_{xz} - M_{Ty} \\ N = \dot{r}I_z - \dot{p}I_{xz} + pq(I_{yy} - I_{xx}) + qrI_{xz} - M_{Tz} \end{cases} \tag{26}$$

Those aerodynamic coefficients are functions of the time histories of the aircraft state, i.e. angle-of-attack α, angle-of-sideslip β, the aircraft rotation rates p, q, r, as well as the control surface deflections δ_A, δ_R, δ_E. The functional relationship between the aerodynamic coefficients and the state variables are expressed in terms of Taylor's series expansions about a reference state [48]. For more technical details, see Appendix.

$$\begin{aligned} C_x &= C_{\alpha x} + C_{qx} + C_{\delta x} + C_{x7}\mathrm{F_T} & C_l &= C_{\beta l} + C_{pl} + C_{rl} + C_{\delta l} + C_{l7}(y_{\mathrm{gc}} - y_\mathrm{s})\mathrm{F_T} \\ C_y &= C_{\beta y} + C_{py} + C_{ry} + C_{\delta y} + C_{y8}\beta\mathrm{F_T} & C_m &= C_{\alpha m} + C_{\dot{\alpha} m} + C_{qm} + C_{\delta m} + C_{m7}\alpha F_T \\ C_z &= C_{\alpha z} + C_{qz} + C_{\delta z} + C_{z7}\alpha\mathrm{F_T} & C_n &= C_{\beta n} + C_{pn} + C_{rn} + C_{\delta n} + C_{n7}(z_{\mathrm{gc}} - z_\mathrm{s})\mathrm{F_T} \end{aligned} \tag{27}$$

From equations (25), (26), (27) the aerodynamic coefficients can be expressed in vector and matrix form as $Y = A\Theta$ where $Y \in \Re^{6\times1}$ is the vector of the variable need to be filled from experimental data, $A \in \Re^{6\times32}$ is the dimensional matrix of explanatory variables, and $\Theta \in \Re^{32\times1}$ is the vector of system parameters, where $Y = [C_x, C_y, C_z, C_l, C_m, C_n]^T \in \Re^{6\times1}$ and $A = [A_1, \dots A_6]^T \in \Re^{6\times32}$ and $\Theta = [\Theta_1, \dots, \Theta_6]^T \in \Re^{32\times1}$. The detailed form of those variables can be found in [49]. Rewrite the linear model of $Y = A\Theta$ as

$$[A|Y][\Theta^T|-1]^T = 0 \tag{28}$$

The singular value decomposition of $[A|Y]$ is.

$$[A|Y] = U\Sigma V^T = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_{33} \end{bmatrix} \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix}^T \tag{29}$$

where $\sigma_1 \geq \cdots \geq \sigma_{33}$ be the singular values of $[A|Y]$. The total least squares solution is given by $\hat{\Theta}_{\text{TLSE}} = -V_{12}V_{22}^{-1}$. This solution exists if and only if V_{22} is non-singular. In addition, it is unique if and only if $\sigma_n \neq \sigma_{n+1}$. In what we give the basic total least squares algorithm follows.

Algorithm: Basic total least squares algorithm
Input: $A \in \Re^{m\times n}$ and $Y \in \Re^{m\times d}$
Compute the SVD decomposition: $[A\|Y] = U\Sigma V^T$
if V_{22} is nonsingular then set $\hat{\Theta}_{\text{TLSE}} = -V_{12}V_{22}^{-1}$;
else Print: the problem has no solution and stop.
end
Output: $\hat{\Theta}_{\text{TLSE}}$ a TLS solution of $Y = A\hat{\Theta}_{TLSE}$

To ensure identifiability (a partial model was initially taken from official technical reports), the aircraft was excited using diverse nonlinear maneuvers within the SP-ACS environment, assuming zero-mean, bounded noise on both inputs and outputs. The TLSE method was preferred over RLS and Kalman filtering due to its robustness against noise and rank-deficient data, which often occur during fast or stalled flight conditions. Mean aerodynamic coefficients were then identified under varied simulation scenarios using SP-ACS and PPjoy (Table 4), providing a stable approximation across flight regimes and enhancing control robustness when real-time data are limited [50].

Table 4. Soukhoï-30 Aerodynamic Coeff-derivatives

$C_{x0} = 0.081$	$C_{y0} = 0.061$	$C_{z0} = 1.810$	$C_{l1} = 0.082$	$C_{m0} = 0.120$	$C_{n1} = 0.421$
$C_{x1} = 0.503$	$C_{y1} = 0.042$	$C_{z1} = 0.210$	$C_{l2} = 0.132$	$C_{m1} = 0.112$	$C_{n2} = 0.051$
$C_{x2} = 0.912$	$C_{y3} = 0.251$	$C_{z3} = 0.415$	$C_{l3} = 0.097$	$C_{m2} = 0.114$	$C_{n3} = 0.045$
$C_{x3} = 0.841$	$C_{y4} = 0.183$	$C_{z5} = 0.020$	$C_{l4} = 0.251$	$C_{m3} = 0.115$	$C_{n4} = 0.110$
$C_{x4} = 0.001$	$C_{y6} = 0.061$	$C_{z7} = 0.123$	$C_{l5} = 0.024$	$C_{m4} = 0.132$	$C_{n5} = 0.220$
$C_{x5} = 0.821$	$C_{y7} = 0.450$				
$C_{x7} = 0.930$					

4. New Asymmetric Surface Control

To Aircraft control surfaces are classified into primary (ailerons, rudder, elevator) and secondary (tabs, flaps, spoilers, slats) types. Some aircraft use symmetric aileron/elevator deflections, while others adopt asymmetric (differential) ones, where the upward-moving surface deflects more than the downward, reducing rudder use during turns [42], [51]. Advanced systems may feature independent left/right surfaces such as $(\delta_{LA}, \delta_{RA})$, $(\delta_{LE}, \delta_{RE})$, and δ_R, operating symmetrically or differentially, especially in fly-by-wire systems. Some aircraft can disconnect jammed surfaces or use separate hydraulics to maintain control under failure [52]. While split surfaces can be implemented on any aircraft, they are mainly used in advanced designs where added

maneuverability, redundancy, or aerodynamic needs justify the complexity. Design references like Raymer's *Aircraft Design* discuss these configurations and their roles.

4.1. Control of Non-Conventional Surface Aircrafts

The previous controlled aerodynamic-propulsive forces and moments can be written as: $\mathrm{F_{AP}} = \mathrm{F_1} + \Pi_\mathrm{F}\delta_\mathrm{asy}$, $\mathrm{M_{AP}} = \mathrm{M_1} + \Pi_\mathrm{M}\delta_\mathrm{asy}$ with $\delta_\mathrm{asy} = [\delta_{La}\ \delta_{Ra}\ \delta_r\ \delta_{Le}\ \delta_{Re}\ \delta_{th}]^T \in \Re^{6\times1}$ is the control signal, $\Pi_\mathrm{F} \in \Re^{3\times6}$ and $\Pi_\mathrm{M} \in \Re^{3\times6}$ are matrices of aero-propulsive coefficients due to the controls. Here is a compact form of dynamic model:

$$\frac{d}{dt}\begin{bmatrix}\mathrm{v}\\ \omega\end{bmatrix} = \begin{bmatrix}(\mathrm{F_1}+\mathrm{F_G})/m - \omega\times\mathrm{v}\\ \mathbb{I}_G^{-1}\left(\mathrm{M_1} - \omega\times(\mathbb{I}_G.\omega)\right)\end{bmatrix} + \begin{bmatrix}\Pi_\mathrm{F}/m\\ \mathbb{I}_G^{-1}\Pi_\mathrm{M}\end{bmatrix}\delta_\mathrm{asy} \text{ and } \frac{d}{dt}\begin{bmatrix}\mathrm{r}\\ \vartheta\end{bmatrix} = \begin{bmatrix}\mathrm{T_1}(\vartheta) & 0\\ 0 & \mathrm{T_0}(\vartheta)\end{bmatrix}\begin{bmatrix}\mathrm{v}\\ \omega\end{bmatrix} \quad (30)$$

If we let $\mathrm{x}^T = [\mathrm{x}_1^T\ \ \mathrm{x}_2^T] \in \Re^{12\times1}$ be the state vector of the above dynamic system with $\mathrm{x}_1^T = [\mathrm{v}^T\ \omega^T]$ and $\mathrm{x}_2^T = [\mathrm{r}^T\ \vartheta^T]$ then $\dot{\mathrm{x}}_1 = \mathrm{f_{asy1}}(\mathrm{x}_1,\mathrm{x}_2) + \Pi\delta_\mathrm{asy}$; $\dot{\mathrm{x}}_2 = \mathrm{f_{asy2}}(\mathrm{x}_1,\mathrm{x}_2)$ where.

$$\mathrm{f_{asy1}}(\mathrm{x}) = \begin{bmatrix}(\mathrm{F_1}+\mathrm{F_G})/m - \omega\times\mathrm{v}\\ \mathbb{I}_G^{-1}\left(\mathrm{M_1} - \omega\times(\mathbb{I}_G.\omega)\right)\end{bmatrix};\quad \mathrm{f_{asy2}}(\mathrm{x}) = \begin{bmatrix}\mathrm{T_1}(\vartheta)\mathrm{v}\\ \mathrm{T_0}(\vartheta)\omega\end{bmatrix} \text{ and } \Pi = \begin{bmatrix}\Pi_\mathrm{F}/m\\ \mathbb{I}_G^{-1}\Pi_\mathrm{M}\end{bmatrix} \in \Re^{6\times6}$$

Notice that: $\mathrm{x}_1(t) = \mathrm{T}^{-1}(\vartheta)\dot{\mathrm{x}}_2(t)$ where $\mathrm{T}(\vartheta) = \mathrm{blkdiag}([\mathrm{T_1},\ \mathrm{T_0}])$, so if we propose the linear controller $\dot{\mathrm{x}}_2(t) = \dot{\mathrm{x}}_{d2}(t) + \mathrm{K_2e_2}(t)$ with an error vector $\mathrm{e}_2(t) = \mathrm{x}_{d2}(t) - \mathrm{x}_2(t)$. Therefore, the commanded signal x_{c1} is $\mathrm{x}_{c1}(t) = \mathrm{T}^{-1}(\vartheta)(\dot{\mathrm{x}}_{d2} + \mathrm{K_2e_2})$. The inner loop dynamic inversion control is given by $\delta_\mathrm{asy}(t) = \Pi^{-1}\left(\dot{\mathrm{x}}_1(t) - \mathrm{f_1}(\mathrm{x}_1,\mathrm{x}_2)\right)$ and we impose the correction $\dot{\mathrm{x}}_1(t) = \dot{\mathrm{x}}_{d1}(t) + \mathrm{K_1e_1}(t)$ with an error $\mathrm{e}_1(t) = \mathrm{x}_{d1}(t) - \mathrm{x}_1(t)$ and let $\mathrm{x}_{d1}(t) = \mathrm{x}_{c1}(t)$ then.

$$\mathrm{x}_{c1}(t) = \mathrm{T}^{-1}(\vartheta)\left(\dot{\mathrm{x}}_{d2}(t) + \mathrm{K_2e_2}(t)\right) \text{ and } \delta_\mathrm{asy}(t) = \Pi^{-1}\left[\dot{\mathrm{x}}_{c1}(t) + \mathrm{K_1e_1}(t) - \mathrm{f_{asy1}}(\mathrm{x}_1,\mathrm{x}_2)\right]$$

Fig. 10 illustrate the general structure of the asymmetric autopilot-controller.

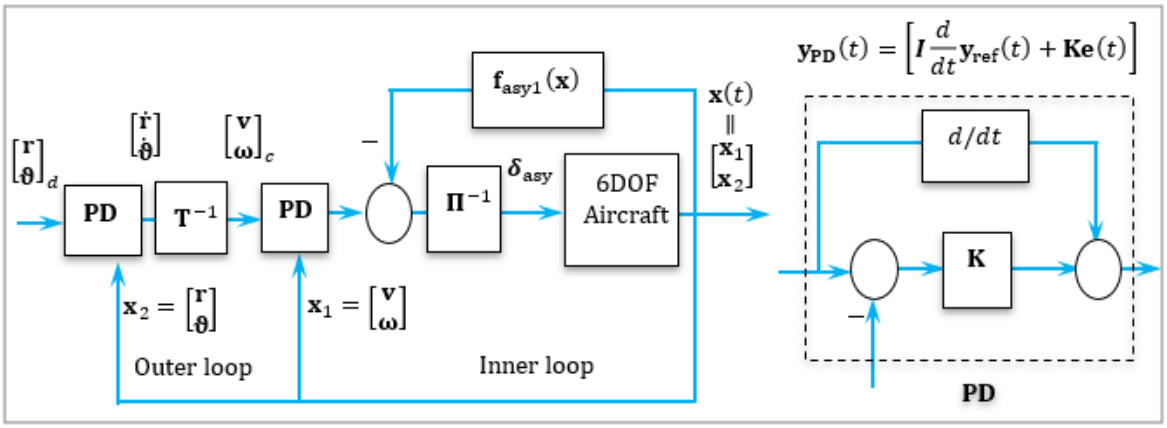

Fig. 10. Structural schematic diagram of the non-adaptive asymmetric autopilot-controller

4.2. Intelligent Asymmetric Control

Now, we propose a robust nonlinear control strategy for a fighter aircraft, combining dynamic inversion with neural networks to enhance performance under uncertain conditions and un-modeled dynamics. By incorporating NN, the controller adapts to variations in the aircraft's behavior, ensuring precise control across a wide flight envelope. The method focuses on achieving improved maneuverability through asymmetric surface control, enabling finer adjustments in roll, pitch, and yaw. Consider the uncertain dynamic model of an aircraft (in the non-affine form):

$$\frac{d}{dt}\begin{bmatrix}\mathrm{v}\\ \omega\end{bmatrix} = \begin{bmatrix}(\mathrm{F_1}+\mathrm{F_G})/m - \omega\times\mathrm{v}\\ \mathbb{I}_G^{-1}\left(\mathrm{M_1} - \omega\times(\mathbb{I}_G.\omega)\right)\end{bmatrix} + \begin{bmatrix}\Pi_\mathrm{F}/m\\ \mathbb{I}_G^{-1}\Pi_\mathrm{M}\end{bmatrix}\delta_\mathrm{asy} + \Delta_1;\quad \frac{d}{dt}\begin{bmatrix}\mathrm{r}\\ \vartheta\end{bmatrix} = \begin{bmatrix}\mathrm{T_1} & 0\\ 0 & \mathrm{T_0}\end{bmatrix}\begin{bmatrix}\mathrm{v}\\ \omega\end{bmatrix} + \Delta_2 \quad (31)$$

where $\mathrm{v}, \omega, \mathrm{r}, \vartheta \in \Re^3$, $\delta \in \Re^6$, $\Delta_1 \in \Re^6$ and $\Delta_2 \in \Re^6$ stands for model uncertainties and external disturbances. Defining the desired references $\mathrm{x}_{1c}^T = [\mathrm{v}_c^T \ \omega_c^T]$, and $\mathrm{x}_{2d}^T = [\mathrm{r}_d^T \ \vartheta_d^T]$ the tracking error vectors are $\mathrm{e}_1(t) = \mathrm{x}_{1c}(t) - \mathrm{x}_1(t)$, and $\mathrm{e}_2(t) = \mathrm{x}_{2d}(t) - \mathrm{x}_2(t)$. Now, the control commands can be defined as.

$$\begin{aligned} \delta_{\text{asy}} &= \begin{bmatrix} \Pi_{\mathrm{F}}/m \\ \mathbb{I}_G^{-1}\Pi_{\mathrm{M}} \end{bmatrix}^{-1} \left[\frac{d}{dt} \begin{bmatrix} \mathrm{v} \\ \omega \end{bmatrix}_c + \mathrm{K}_1 \begin{bmatrix} \mathrm{v}_c - \mathrm{v} \\ \omega_c - \omega \end{bmatrix} - \mathrm{f}_{\text{asy1}}(\mathrm{x}_1, \mathrm{x}_2) - \Delta_1 \right] \\ \begin{bmatrix} \mathrm{v} \\ \omega \end{bmatrix}_c &= \begin{bmatrix} \mathrm{T}_1(\vartheta) & 0 \\ 0 & \mathrm{T}_0(\vartheta) \end{bmatrix}^{-1} \left\{ \frac{d}{dt} \begin{bmatrix} \mathrm{r}_d \\ \vartheta_d \end{bmatrix} + \mathrm{K}_2 \begin{bmatrix} \mathrm{r}_d - \mathrm{r} \\ \vartheta_d - \vartheta \end{bmatrix} - \Delta_2 \right\} \end{aligned} \tag{32}$$

With

$$\mathrm{f}_{\text{asy1}}(\mathrm{x}_1, \mathrm{x}_2) = \begin{bmatrix} (\mathrm{F}_1 + \mathrm{F}_{\mathrm{G}})/m - \omega \times \mathrm{v} \\ \mathbb{I}_G^{-1}\left(\mathrm{M}_1 - \omega \times (\mathbb{I}_G.\omega)\right) \end{bmatrix}; \quad \frac{d\mathrm{W}_1}{dt} = \Gamma_1 \mu_1 \mathrm{e}_1^T; \quad \frac{d\mathrm{W}_2}{dt} = \Gamma_2 \mu_2 \mathrm{e}_2^T; \quad \begin{aligned} \Delta_1 &= \mathrm{W}_1^T \mu_1 \\ \Delta_2 &= \mathrm{W}_2^T \mu_2 \end{aligned}$$

Recursive algorithms can be used to estimate parameters and states in a wide variety of models. The recursive least square RLS approach begins with the following difference equation model, $\mathrm{n}^T(k) = \mu^T(k)\mathrm{W}(k) + \mathrm{e}(k)$ here we can estimate the model by minimizes the errors of the model. In this setting, we will select the cost, $\sum_{k=1}^{N} \mathrm{e}^T(k)\mathrm{e}(k)/N$, minimizing this equation will yield tp $\mathrm{W} = \left[\sum_{k=1}^{N} \mu(k)\mu^T(k)\right]^{-1}\left[\sum_{k=1}^{N} \mu(k)\mathrm{y}^T(k)\right]$. To avoid matrix inversion we use the matrix inversion lemma, to get the well-known recursive least square algorithm.

(1) $\mathrm{W}(k+1) = \mathrm{W}(k) + \mathrm{K}(k)\mathrm{e}(k)$ (3) $\mathrm{e}(k) = \mathrm{n}^T(k) - \mu^T(k)\mathrm{W}(k)$

(2) $\mathrm{P}(k+1) = \left(\mathrm{I} - \mathrm{K}(k)\mu^T(k)\right)\mathrm{P}(k)$ (4) $\mathrm{K}(k) = \mathrm{P}(k)\mu(k)/\left(1 + \mu^T(k)\mathrm{P}(k)\mu(k)\right)$

An enhanced version of the RLS is the forgetting factor method, used improve the adaptability when tracking time-varying systems [36], [37]. Fig. 11 illustrate the detailed structure of the asymmetric autopilot-controller.

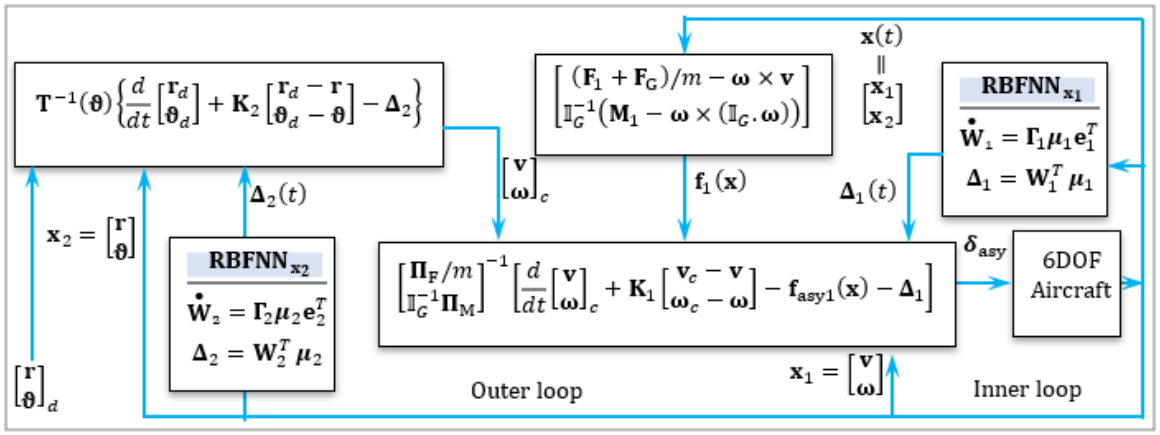

Fig. 11. Detailed structural diagram of an asymmetric autopilot-controller function

5. Practical Adaptive Control Strategy

A practical control strategy for fighter jets targets direct control of angle of attack (α) and sideslip angle (β), improving agility and precision in combat. Unlike conventional axis-based methods, this approach directly modulates aerodynamic forces, and when combined with NDI and NNs, boosts maneuverability and response. The nonlinear 6-DOF equations of motion over a flat Earth, expressed in the body-fixed frame, are given by the following differential equations [40], [41].

$$\left\{\begin{array}{ll} \dot{V}=\frac{F_x}{m}\mathrm{c}_\alpha\mathrm{c}_\beta+\frac{F_y}{m}\mathrm{s}_\beta+\frac{F_z}{m}\mathrm{s}_\alpha\mathrm{c}_\beta-\mathrm{gf}_1 & \dot{\phi}=p+\frac{\mathrm{s}_\theta}{\mathrm{c}_\theta}\left[q\mathrm{s}_\phi+r\mathrm{c}_\phi\right] \\ \dot{\alpha}=q-\frac{\mathrm{s}_\beta}{\mathrm{c}_\beta}\left[p\mathrm{c}_\alpha+r\mathrm{s}_\alpha\right]-\frac{F_x}{m}\frac{\mathrm{s}_\alpha}{V\mathrm{c}_\beta}+\frac{F_z}{m}\frac{\mathrm{c}_\alpha}{V\mathrm{c}_\beta}+\frac{\mathrm{gf}_2}{V\mathrm{c}_\beta} & \dot{\theta}=q\mathrm{c}_\phi-r\mathrm{s}_\phi \\ & \dot{p}=a_1L+a_2N-[a_3p+a_4r]q \\ \dot{\beta}=p\mathrm{s}_\alpha-r\mathrm{c}_\alpha-\frac{F_x}{m}\mathrm{c}_\alpha\mathrm{s}_\beta+\frac{F_y}{m}\mathrm{c}_\beta-\frac{F_z}{m}\mathrm{s}_\alpha\mathrm{s}_\beta+\frac{\mathrm{gf}_3}{V} & \dot{q}=a_5M+a_6[r^2-p^2]-a_7pr \\ & \dot{r}=a_2L+a_8N+[a_3r+a_9p]q \end{array}\right\} \quad (33)$$

where $a_1=I_{zz}/[I_{xx}I_{zz}-I_{xz}^2], a_2=I_{xz}/[I_{xx}I_{zz}-I_{xz}^2],\ a_3=I_{xz}\left[I_{yy}-I_{xx}-I_{zz}\right]/[I_{xx}I_{zz}-I_{xz}^2]$, $a_4=\left[I_{xz}^2+I_{zz}\left(I_{zz}-I_{yy}\right)\right]/[I_{xx}I_{zz}-I_{xz}^2],\ a_5=1/I_{yy},\ a_6=I_{xz}/I_{yy},\ a_7=[I_{xx}-I_{zz}]/I_{yy},\ a_8=I_{xx}/[I_{xx}I_{zz}-I_{xz}^2], a_8=\left[I_{xz}^2+I_{xx}\left(I_{xx}-I_{yy}\right)\right]/[I_{xx}I_{zz}-I_{xz}^2]$,

$$\begin{bmatrix}\mathrm{f}_1\\ \mathrm{f}_2\\ \mathrm{f}_3\end{bmatrix}=\begin{bmatrix}\mathrm{s}_\theta\mathrm{c}_\alpha\mathrm{c}_\beta-\mathrm{s}_\theta\mathrm{s}_\phi\mathrm{s}_\beta-\mathrm{c}_\theta\mathrm{c}_\phi\mathrm{s}_\alpha\mathrm{c}_\beta\\ \mathrm{c}_\theta\mathrm{c}_\alpha\mathrm{c}_\phi+\mathrm{s}_\theta\mathrm{s}_\alpha\\ \mathrm{s}_\theta\mathrm{c}_\alpha\mathrm{s}_\beta+\mathrm{c}_\theta\mathrm{s}_\phi\mathrm{c}_\beta-\mathrm{c}_\theta\mathrm{c}_\phi\mathrm{s}_\alpha\mathrm{s}_\beta\end{bmatrix};\text{and}\begin{bmatrix}F_x\\ F_y\\ F_z\end{bmatrix}=\begin{bmatrix}-\mathrm{c}_\alpha\mathrm{c}_\beta & -\mathrm{c}_\alpha\mathrm{s}_\beta & \mathrm{s}_\alpha\\ -\mathrm{s}_\beta & \mathrm{c}_\beta & 0\\ -\mathrm{s}_\alpha\mathrm{c}_\beta & -\mathrm{s}_\alpha\mathrm{s}_\beta & -\mathrm{c}_\alpha\end{bmatrix}\begin{bmatrix}F_D\\ F_Y\\ F_L\end{bmatrix}+\begin{bmatrix}F_T\\ 0\\ 0\end{bmatrix}$$

with: F_x, F_y, and F_z are guidance forces about the body-fixed frame [N]; α is the angle of attack [rad] or [deg]; β is the side slip angle [rad] or [deg]; θ and ϕ are Euler angles [rad] or [deg]; ϕ is the rolling angle; $\dot{\theta}$ is the angular velocity regarding to y axis; F_T means force of the thrust [N]; F_L is the lift force [N]; F_Y is the side force [N]; F_D is the drag force [N]; p, q, and r are components of airplane's angular velocity regarding x, y, z body axes [rad/s]; I_{xx}, I_{yy}, I_{zz} and I_{xz} are moments of inertia [kg/m^2]; L, M, and N are aerodynamic rolling, pitching, and yawing moment; m is mass [kg]; g is the acceleration due to gravity; V is the magnitude of the velocity.

The aerodynamic forces F_D, F_Y, F_L (drag, side, lift) and moments L, M, N (rolling, pitching, yawing) used in the model are given by the following equations:

$$\left\{\begin{array}{l} F_L=Q\mathrm{s}\left(C_{L0}+C_{L\alpha}\alpha+C_{L\dot{\alpha}}\dot{\alpha}+C_{Lq}q+C_{L\delta_e}\delta_e\right)\\ F_Y=Q\mathrm{s}\left(C_{Y\beta}\beta+C_{Yp}p+C_{Yr}r+C_{Y\delta_a}\delta_a+C_{Y\delta_r}\delta_r\right)\\ F_D=Q\mathrm{s}\left(C_{D0}+C_{D\alpha}\alpha+C_{D\beta}\beta+C_{D\delta_e}\delta_e+C_{D\delta_r}\delta_r\right)\\ L=bQ\mathrm{s}\left(C_{L\beta}\beta+C_{Lp}p+C_{Lr}r+C_{L\delta_r}\delta_r+C_{L\delta_a}\delta_a\right)\\ M=cQ\mathrm{s}\left(C_{M0}+C_{M\alpha}\alpha+C_{Mq}q+C_{M\delta_e}\delta_e\right)\\ N=bQ\mathrm{s}\left(C_{N\beta}\beta+C_{Np}p+C_{Nr}r+C_{N\delta a}\delta_a+C_{N\delta r}\delta_r\right)\end{array}\right\} \quad (34)$$

5.1. Intelligent Symmetric Adaptive Practical Control

Classical nonlinear dynamic inversion assumes known dynamics and input-affine systems—conditions rarely met in fault-tolerant flight. Adaptive ANDI addresses this using neural networks or real-time identification [11], [32], [34] and [38], [39]. Here, we propose an ANDI scheme based on a single-layer RBF perceptron and employ a two-timescale strategy: the inner-loop regulates fast attitude rates via $\mathrm{u}=[\delta_a;\delta_e;\delta_r]$, driven by outer-loop targets. By applying NDI to the last three equations in (33), the fast-state dynamics are decoupled and regulated accordingly.

$$\dot{\omega}(t)=\mathrm{f}_\omega(\xi)+A_\omega(\xi)\mathrm{u}(t)+\Delta_\omega(t) \quad (35)$$

where $A_\omega(\xi)$ and $\mathrm{f}_\omega(\xi)=\left[\mathrm{f}_p(\xi)\quad \mathrm{f}_q(\xi)\quad \mathrm{f}_r(\xi)\right]^T$ can be derived from equation (33), and $\omega(t)=[p(t)\ q(t)\ r(t)]^T$. Considering the dynamic inversion, the controller of inner loop yields: $\mathrm{u}(t)=A_\omega^{-1}(\xi)[\dot{\omega}(t)-\mathrm{f}_\omega(\xi)-\Delta_\omega(t)];\ \Delta_\omega(t)=\mathrm{W}_\omega^T\mu_\omega;$ and $d\mathrm{W}_\omega/dt=\Gamma_\omega\mu_\omega\mathrm{e}_\omega^T$.

The angular rates are defined by: $\dot{\omega}(t)=\dot{\omega}_c(t)+\mathrm{K}_\omega\mathrm{e}_\omega$ where $\mathrm{e}_\omega(t)=\omega_c(t)-\omega(t)$ and $\mathrm{K}_\omega=\mathrm{diag}\left[K_p\ \ K_q\ \ K_r\right]$ is the inner-loop gain chosen by the designer to obtain desire performance. The outer loop controls slow states (α,β,ϕ), assuming fast states respond instantaneously and

surface deflections don't affect outer-loop dynamics. Its output serves as the inner-loop command. The slow-state evolution follows a differential equation:

$$\dot{\eta}(t) = \mathrm{f}_\eta(\xi) + A_\eta(\xi)\omega(t) + A_o(\xi)\mathrm{u}(t) + \Delta_\eta(t) \tag{36}$$

while $A_\eta(\xi)$, $A_o(\xi)$ and $\mathrm{f}_\eta(\xi) = \begin{bmatrix}\mathrm{f}_\alpha(\xi) & \mathrm{f}_\beta(\xi) & \mathrm{f}_\phi(\xi)\end{bmatrix}^T$ can be derived from the model, and $\eta(t) = [\alpha(t) \quad \beta(t) \quad \phi(t)]^T$. Considering the DNI method, the controller of outer loop yields: $\omega(t) = A_\eta^{-1}(\xi)\left[\dot{\eta}(t) - \mathrm{f}_\eta(\xi) - A_o(\xi)\mathrm{u}(t) - \Delta_\eta(t)\right]$. Since $\|A_o\mathrm{u}(t)\| \ll \left\|\dot{\eta}(t) - \mathrm{f}_\eta(\xi) - \Delta_\eta(t)\right\|$ then the small term $A_o(\xi)\mathrm{u}(t)$ is neglected. The output of outer loop is derived as:

$$\omega_c(t) = A_\eta^{-1}(\xi)\left[\dot{\eta}(t) - \mathrm{f}_\eta(\xi) - \Delta_\eta(t)\right]; \quad \text{with} \quad \Delta_\eta(t) = \mathrm{W}_\eta^T\,\mu_\eta; \quad d\mathrm{W}_\eta/dt = \Gamma_\eta\mu_\eta\mathrm{e}_\eta^T \tag{37}$$

The angular derivatives $\dot{\eta}(t)$ are given by: $\dot{\eta}(t) = \dot{\eta}_d(t) + \mathrm{K}_\eta\mathrm{e}_\eta(t)$ where $\mathrm{e}_\eta(t) = \eta_d(t) - \eta(t)$ and $\mathrm{K}_\eta = \mathrm{diag}\left[K_\alpha \quad K_\beta \quad K_\phi\right]$, K_α, K_β, and K_ϕ are outer loop control gains, which are chosen by the designer to obtain desired performance; α_d, β_d, and ϕ_d are guidance system commands. The DI-based velocity controller computes the thrust command by substituting $\dot{V}$ with $K_V e_V$ and using the desired thrust F_{Td}. By dividing F_{Td} with maximum thrust $F_{T\max}$, the throttle input δ_{TH} is then given by:

$$\delta_{TH} = \left[mK_V e_V + F_D\mathrm{c}_\beta - F_Y\mathrm{s}_\beta + mg\mathrm{f}_1 - \Delta_V\right]/F_{T\max}\mathrm{c}_\alpha\mathrm{c}_\beta \tag{38}$$

With $\mathrm{f}_1 = \left[\mathrm{s}_\theta\mathrm{c}_\alpha\mathrm{c}_\beta - \mathrm{s}_\theta\mathrm{s}_\phi\mathrm{s}_\beta - \mathrm{c}_\theta\mathrm{c}_\phi\mathrm{s}_\alpha\mathrm{c}_\beta\right]$; $\quad \Delta_V(t) = \mathrm{W}_\mathrm{V}^T\,\mu_\mathrm{V}$; $\quad d\mathrm{W}_\mathrm{V}/dt = \Gamma_\mathrm{V}\mu_\mathrm{V}\mathrm{e}_V^T$

where: K_V: a control gain and, $e_V = V_d - V$: a relative speed error.

5.2. Guidance Processor Design (A Proposed Pure Pursuit Guidance Law)

Pure pursuit aligns the follower's velocity vector with the line-of-sight (LOS) to a target or virtual leader. For this, velocity v must satisfy: $\mathrm{v} = \|\mathrm{v}\|\mathrm{u}_r$, where $\mathrm{u}_r(t) = \mathrm{r}_0(t)/\|\mathrm{r}_0(t)\|$ is the unit LOS vector. The follower's acceleration (fighter aircraft acceleration) a_F is derived accordingly from classical pure pursuit navigation.

$$\mathrm{a}_\mathrm{F}(t) = \frac{d\mathrm{v}}{dt} = \frac{d\|\mathrm{v}\|}{dt}\mathrm{u}_r(t) + \|\mathrm{v}\|\frac{d\mathrm{u}_r}{dt} = \frac{dV}{dt}\frac{\mathrm{r}_0}{r} + \frac{V}{r^2}\left[r\frac{d\mathrm{r}_0}{dt} - \mathrm{r}_0\frac{dr}{dt}\right] \tag{39}$$

where $V(t) = \|\mathrm{v}\|$, $r(t) = \|\mathrm{r}_0\|$ and let we define $\mathrm{v}_c(t) = -d\mathrm{r}_0/dt$ be the closing velocity and $V_\mathrm{c}(t) = dr/dt = \|\mathrm{v}_c\|$ then.

$$\mathrm{a}_\mathrm{F}(t) = \frac{dV}{dt}\frac{\mathrm{r}_0}{r} + \frac{V}{r^2}\left[r\frac{d\mathrm{r}_0}{dt} - \mathrm{r}_0\frac{dr}{dt}\right] = \left\{\frac{1}{r(t)}\frac{dV}{dt} - \frac{V(t).V_\mathrm{c}(t)}{r^2(t)}\right\}\mathrm{r}_0(t) - \frac{V(t)}{r(t)}\mathrm{v}_c(t) \tag{40}$$

Here, there is some complexity in the formula of a_F because in computing a_F we need first to evaluate dV/dt which is not yet computed, so in order to overcome such problem we assume that the term dV/dt is proportional to the target acceleration: $dV/dt = K\|\mathrm{a}_\mathrm{T}\|$ with K= proportionality constant.

$$\frac{dV}{dt} = K\|\mathrm{a}_\mathrm{T}\| \quad \Longrightarrow \quad \mathrm{a}_\mathrm{F}(t) = \left\{\frac{K}{r(t)}\|\mathrm{a}_\mathrm{T}(t)\| - \frac{V(t).V_\mathrm{c}(t)}{r^2(t)}\right\}\mathrm{r}_0(t) - \frac{V(t)}{r(t)}\mathrm{v}_c(t) \tag{41}$$

To improve the effectiveness of the proposed commanded acceleration, we should implement a guidance law that is more robust to uncertainties and ensures stability. For such purpose let we select the manifold $\mathrm{s}(\xi) = \mathrm{r}_o \times \mathrm{v} \in \Re^{n\times 1}$ as a sliding surface, then the derivative of this surface with respect to time is $\dot{\mathrm{s}}(\xi) = \dot{\mathrm{r}}_o \times \mathrm{v} + \mathrm{r}_o \times \mathrm{a}_\mathrm{F}$, but the pure pursuit guidance impose the condition $\mathrm{v} = (V/r)\mathrm{r}_o$ therefore,

$$\dot{s} = \frac{dr_o}{dt} \times v + r_o \times a_F = \frac{V}{r}\frac{dr_o}{dt} \times r_o + r_o \times a_F = r_o \times \left[a_F - \frac{V}{r}\frac{dr_o}{dt}\right] = r_o \times \left[a_F + \frac{V}{r}v_c\right] \quad (42)$$

The convergence is guaranteed if the states of the dynamic system approach the sliding surface. In order to obtain this condition, we defined a positive definite function: $E(\xi) = s^T(\xi).s(\xi)/2 = \|s(\xi)\|^2/2$ as a *Lyapunov candidate* for which, its time derivative is given by $\dot{E}(\xi) = s^T(\xi).\dot{s}(\xi)$ and by Lyapunov theorem, a system is said to be stable iff:

$$\dot{E}(\xi) \leq 0 \quad \text{so} \quad s^T(\xi).\dot{s}(\xi) \leq 0 \quad (43)$$

If we choose the sliding surface $s(\xi)$ such that

$$\dot{s}(\xi) = -K.\text{sign}\big(s(\xi)\big) \in \Re^{n\times 1} \quad (44)$$

with positive definite matrix $K \in \Re^{n\times n}$ then the time derivative of $E(\xi)$ is

$$\dot{E}(\xi) = -s^T(\xi).K.\text{sign}\big(s(\xi)\big) \leq 0 \quad (45)$$

This is always satisfied see [8]. Now, we select the input acceleration a_F in order to force $\dot{s}(\xi)$ to be in the predicted form: $\dot{s}(\xi) = -K_2\text{sign}\big(s(\xi)\big)$. First of all, we assume that a_F is compose of $a_F = a_{F1} + a_{F2} + a_{F3}$ and according to equation (42) we have

$$\dot{s}(\xi) = r_o \times \left\{a_F + \frac{v}{r}v_c\right\} = -K_2\text{sign}\big(s(\xi)\big) \quad (46)$$

Now the term a_{F1} is selected to cancel $(V\dot{r}_o/r)$ and the term a_{F2} is selected to nullify the cross product, $a_{F2} = K_1.r_o$ and a_{F3} is selected as $a_{F3} = [K_2.r_o/r^2] \times \text{sign}\{s(\xi)\}$. Collect the above accelerations we obtain.

$$a_F = \left\{K_1.r_o - \left(\frac{V}{r}\right)v_c\right\} + \left(\frac{K_2}{r^2}\right)r_o \times \text{sign}\big(s(\xi)\big) \quad (47)$$

Notice that this acceleration can be decomposed of two essential terms, namely the equivalent control law a_{equ} and the discontinuous control law a_{dis}:

$$a_{equ} = K_1.r_o - \left(\frac{V}{r}\right)v_c;\ a_{dis} = \left(\frac{K_2}{r^2}\right)r_o \times \text{sign}\big(s(\xi)\big) \quad (48)$$

where a_{equ} is responsible for attracting the response to reach the sliding surface and a_{dis} is responsible for holding it there and to remain within a certain limit and don't leave. To enhance the convergence of the proposed engagement we put the magnitude of a_{dis} to grow exponentially

$$a_{cE} = \left\{K_1.r_o + \left(\frac{V}{r}\right)\frac{dr_o}{dt}\right\} - \left(\frac{K_2}{r^2}\right)\|s\|^{K_3}\text{sign}(s) \times r_o \quad (49)$$

To align inertial guidance commands with the aircraft body frame, a rotation matrix is applied: $a_c = T_1^T a_{cE}$. We define a_c by its coordinates $a_c = A_{cx}i_B + A_{cy}j_B + A_{cz}k_B$. In aircraft guidance, the angle of attack α and sideslip angle β govern longitudinal and lateral accelerations, respectively, while the bank angle ϕ induces bank-to-turn (BTT) acceleration. To minimize $e_A = a_c - a = [A_{xc},\ A_{yc},\ A_{zc}] - [A_x,\ A_y,\ A_z] = [e_{Ax}, e_{Ay}, e_{Az}]$, the attitude control inputs $(\phi_c, \beta_c, \alpha_c)$ must be computed accordingly, via the controller:

$$\alpha_c = \mathcal{C}_0(e_{Az});\ \phi_c = \mathcal{C}_0(e_{Ax}); \beta_c = \mathcal{C}_0\big(e_{Ay}\big) \quad (50)$$

If we let $\mathcal{C}_0 =$ PID then, we can summarize the structure of the proposed controllers as:

$$\begin{aligned}
&\mathcal{C}_0:\ \alpha_c = \mathrm{PID}_\alpha(e_{Az});\quad \phi_c = \mathrm{PID}_\phi(e_{Ax});\quad \beta_c = \mathrm{PID}_\beta(e_{Ay})\\
&\mathcal{C}_1: \mathrm{u}(t) = A_\omega^{-1}(\xi)[\dot{\omega}_c(t) + \mathrm{K}_\omega \mathrm{e}_\omega(t) - \mathrm{f}_\omega(\xi) - \Delta_\omega(t)]\\
&\mathcal{C}_2: \omega_c(t) = A_\eta^{-1}(\xi)\left[\dot{\eta}_d(t) + \mathrm{K}_\eta \mathrm{e}_\eta(t) - \mathrm{f}_\eta(\xi) - \Delta_\eta(t)\right]\\
&\mathcal{C}_3: \delta_{TH} = \frac{1}{F_{T\max}\mathrm{c}_\alpha \mathrm{c}_\beta}\left[mK_V e_V + F_D \mathrm{c}_\beta - F_Y \mathrm{s}_\beta + mg\mathrm{f}_1 - \Delta_V\right]
\end{aligned} \tag{51}$$

with the following defined variables and parameters.

$$\mathrm{f}_1 = \left[\mathrm{s}_\theta \mathrm{c}_\alpha \mathrm{c}_\beta - \mathrm{s}_\theta \mathrm{s}_\phi \mathrm{s}_\beta - \mathrm{c}_\theta \mathrm{c}_\phi \mathrm{s}_\alpha \mathrm{c}_\beta\right];\quad \Delta_\omega(t) = \mathrm{W}_\omega^T \mu_\omega;\quad \Delta_\eta(t) = \mathrm{W}_\eta^T \mu_\eta;\quad \Delta_V(t) = \mathrm{W}_\mathrm{V}^T \mu_\mathrm{V}$$

$$\frac{d\mathrm{W}_\omega}{dt} = \Gamma_\omega \mu_\omega \mathrm{e}_\omega^T(t);\qquad \frac{d\mathrm{W}_\eta}{dt} = \Gamma_\eta \mu_\eta \mathrm{e}_\eta^T(t);\qquad \frac{d\mathrm{W}_\mathrm{V}}{dt} = \Gamma_\mathrm{V} \mu_\mathrm{V} \mathrm{e}_V^T(t)$$

$$e_A = \mathrm{a}_c - \mathrm{a} = e_{Ax}\mathrm{i}_B + e_{Ay}\mathrm{j}_B + e_{Az}\mathrm{k}_B;\qquad \mathrm{r}_o = \mathrm{r}_d - \mathrm{r}$$

$$\mathrm{a}_c = \mathrm{T}_1^T\left[\left\{K_1.\mathrm{r}_o + \left(\frac{\mathrm{V}}{r}\right)\frac{d\mathrm{r}_o}{dt}\right\} - \left(\frac{K_2}{r^2}\right)\|\mathrm{s}\|^{K_3}\mathrm{sign}(\mathrm{s}) \times \mathrm{r}_o\right]$$

$$\mathrm{a} = A_x \mathrm{i}_B + A_y \mathrm{j}_B + A_z \mathrm{k}_B;\qquad \text{and}\qquad \mathrm{s} = \mathrm{r}_o \times \mathrm{v}$$

The function $\mathcal{C}_0$ is the guidance controller, which can be classic, adaptive or DI, in this paper a PID controller is used for this purpose (its gains are determined according to the ref [53]). The last step in this flight control design is integration of the guidance law into the control system. The overall view of the designed system can be seen in Fig. 12.

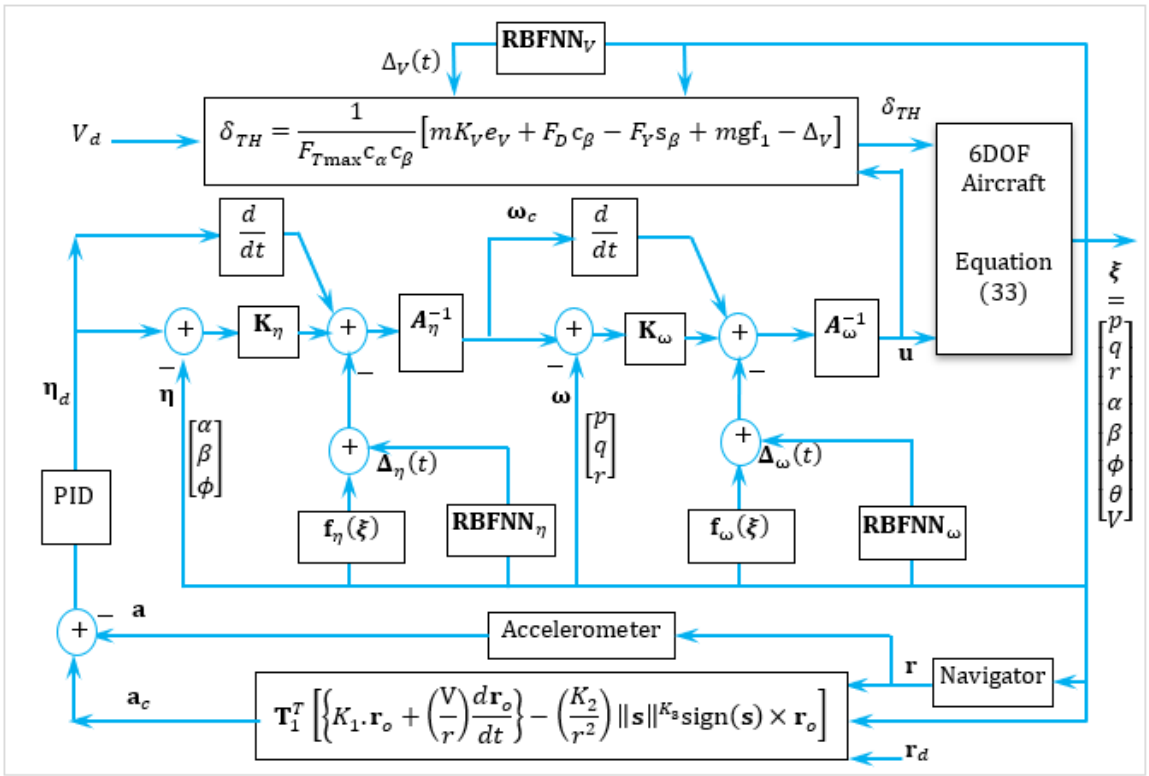

Fig. 12. The integrated guidance and control system diagram

Note: Symmetric control handles position-speed but not direction; asymmetric control improves directionality but ignores speed. To unify benefits, we propose a controller that manages both speed/attack angle, with soft switching to adapt across maneuvers.

The proposed hybrid framework achieves generality because its structure is derived from nonlinear dynamic inversion and adaptive approximation theory, both system-independent formulations. The combination of symmetric and asymmetric loops defines a generalized control

allocation layer applicable to any under-actuated or over-actuated aerial vehicle. The adaptive neural augmentation, based on the universal approximation property of RBF networks, ensures compensation for unknown dynamics without dependence on aircraft-specific models. Furthermore, integrating the sliding-mode-enhanced pursuit law within the same Lyapunov-stable design guarantees convergence for a broad class of 6-DOF systems. Therefore, the approach is not limited to the Su-30 case but can be extended to other aircraft or robotic platforms with similar nonlinear coupling and actuator constraints.

6. Full Simulation Study

This section introduces the SP-ACS framework employed for simulation and validation of the developed autopilot (Fig. 13). The process begins with MATLAB/Simulink-based validation for the Su-30 fighter aircraft, using aerodynamic coefficients estimated via TLSE for controller tuning. Due to the complexity and limited accessibility of real-world testing systems, a fully simulated evaluation is conducted. As in [54], [55], a system-level SP-ACS is developed to support this effort. It consists of three main components: (1) a commercial flight simulator, Microsoft Flight Simulator 2004 (MSFS2004), which offers high-fidelity aircraft dynamics validated by test pilots and supports integration of custom models such as the Su-30 used in this work; (2) an external setup developed in MATLAB/Simulink using Real-Time Windows Target (RTWT) for implementing and executing aircraft control laws in real time; and (3) a communication interface based on shared-memory buffers and the FSUIPC.dll library, which ensures real-time data exchange between MSFS2004 and the external system, allowing seamless transmission of sensor and actuator data. The control loop runs at a fixed sampling period of 1 ms, with hierarchical structuring to ensure that DI and adaptation are executed within real-time constraints.

Fig. 13. Simulation platform of Soukhoï Su-30 control systems

Utilizing Inter-Process Communication (IPC) with a 64KB buffer, the FSUIPC.dll (Flight Simulator Universal Inter- Process Communication) [56]-[60] dynamic link library enables external applications to interact with MSFS2004 by reading and writing data. To manipulate a variable, it's essential to know its address in the FSUIPC table, its data format, and the appropriate communication protocols. For instance, the indicated airspeed is accessed as a signed long integer (S32) at the memory address $0 \times 02BC$. Similarly, throttle position control can be written as unsigned long (S32) at address $0 \times 088C$. Reading the aircraft's current altitude can be written as signed long (S32) at address 0×0574. The elevator deflection $\delta_E (\text{or } \delta_H)$ can be both read and

written as a signed long (S16) at address 0x0BB2. The deflection δ_R can be both read and written as a signed long (S16) at $0 \times 0BC8$. To evaluate system performance, several simplifying assumptions are made: the aircraft is a symmetric, rigid body with constant mass (with uncertainties), and control surfaces respond instantaneously. Gyroscopic and engine effects are neglected, with thrust aligned along the reference line. The Earth is modeled as a flat, inertial frame, and atmospheric conditions are standard and altitude-dependent. No shock waves are considered below the critical Mach number, and drag increases above it. Accelerometers provide actual acceleration, with continuous, globally bounded disturbances. Target data includes direction, distance, and relative velocity, and the controller operates with a small sampling time. The proposed control laws are evaluated through high-fidelity simulations of the Su-30 under dynamic flight conditions and aggressive maneuvers. The strategies—adaptive symmetric, asymmetric, and practical adaptive control—are validated in scenarios like high angles of attack, post-stall maneuvers, and complex trajectory tracking, demonstrating their robustness and applicability. The architecture ensures smooth transitions between control configurations, maintaining performance in varying flight conditions. These simulations verify the control law's ability to enhance maneuverability and robustness under dynamic, uncertain conditions [61]-[64].

6.1. Simulations of Adaptive Symmetric Control

The fighter aircraft is assumed to be controlled by four conventional control inputs: stabilator, rudder, ailerons, and throttle. While throttle is managed separately, the remaining control surfaces are automatically regulated through a closed-loop system. In this subsection, the performance of the designed flight controller, specifically the adaptive symmetric surface control, is evaluated. Simulations are carried out for the Su-30 fighter model under two flight conditions: the first with Mach numbers between $0.7 \leq M \leq 0.9$ at sea level, $h = 0$ [m], and the second with Mach numbers $0.7 \leq M \leq 1.167$ at $h = 7.000$ [m]. Although terms dependent on control surface deflections in the angular rates $\dot{\alpha}$ and $\dot{\beta}$ were disregarded in the design phase, they are included in the un-modeled dynamics to assess the effects of control forces on the aircraft's response. The initial angle of attack is $\alpha_0 = 1.5$ [deg] and the pitch angle is $\theta_0 = 0$ [deg], To generate smooth commands $\phi_d(t)$, $\theta_d(t)$, and $\psi_d(t)$, a virtual reference trajectory is defined by $x_d(t) = 5t^2 \sin(0.2t)$, $y_d(t) = 3t^2 \cos(0.2t)$ and $z_d(t) = 60t^2$ (i.e. Herbst-Like-Maneuver, see Fig. 14). For practical implementation, the control surface deflections are constrained within: $|\delta_a| \leq 30$ [deg], $|\delta_r| \leq 30$ [deg], and $|\delta_e| \leq 30$ [deg]. The closed-loop system's performance, including the full aircraft model and composite control law, is analyzed with a 20% uncertainty band around the nominal values.

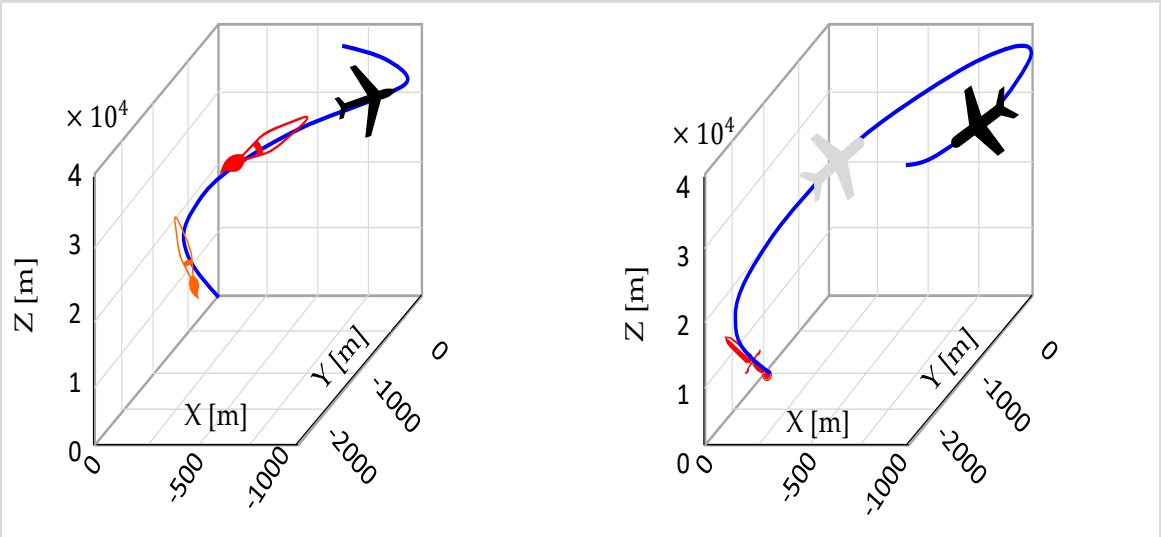

Fig. 14. The 3D trajectory of the fighter (herbst-like-maneuver)

Case I: The terminal values for ϕ_d, θ_d and ψ_d were set as $(\phi_d, \theta_d, \psi_d) = (90, 40, 30)$[deg]. Selected responses are shown in Fig. 15. The results show smooth trajectory tracking, with the steady-state tracking error approaching zero by 2sec. The attack/side-slip angles were calculated using $\alpha = \arctan(w/u)$ and $\beta = \arcsin(v/V)$. The max of side-slip angle β observed during the simulation remained below 0.08^{o}.

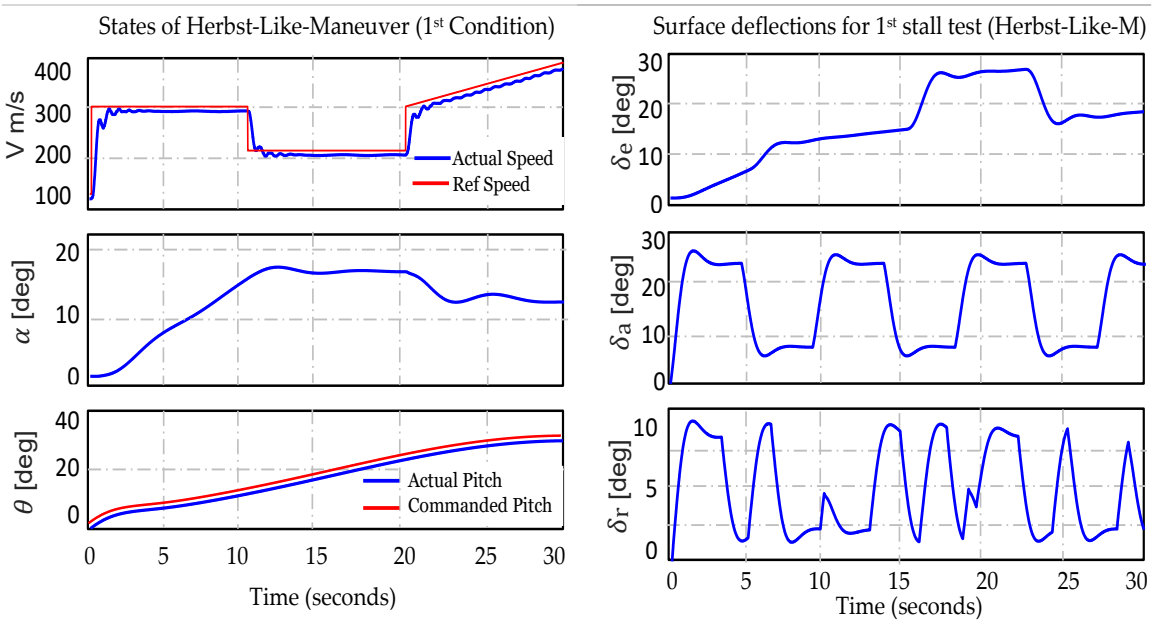

Fig. 15. States and control surface deflections for herbst-like-maneuver (under first stall test)

In Fig. 15 the actual speed V closely tracks its reference, except for a brief period where V dips below the reference. Despite this, the aircraft maintains control as the speed increases significantly. The angle of attack α rises initially, then stabilizes, and begins to decrease slightly towards the end, indicating that the α "AoA" is being reduced to prevent a stall. The pitch angle shows a smooth increase, following the reference. This steady increase indicates a controlled climb. There are no major oscillations, suggesting that the aircraft maintains a stable pitch during the climb phase.

The right-hand side of Fig. 15 displays the deflection angles of the primary control surfaces: elevator (δ_e), ailerons (δ_a), and rudder (δ_r). The elevator deflection gradually increases, stabilizes, then slightly decreases, indicating pitch control. The ailerons, responsible for roll, show periodic deflections, reflecting continuous adjustments to maintain stability or reach a desired bank angle. The rudder, controlling yaw, shows smaller, less frequent oscillations, likely aiding coordinated turns. These deflections reflect active control of flight dynamics, with the elevator managing pitch, ailerons handling roll, and rudder assisting in coordination. The aircraft speed closely tracks the reference, with a brief drop likely due to a maneuver or throttle change. The observed oscillations in the aileron and rudder channels are not structural limit cycles but result from the controller's fast adaptation and sensitivity to noise. These oscillations remain bounded and reflect minor lateral deviations under aerodynamic uncertainty.

Case II: The fighter aircraft was assumed to be flying in the previous conditions with the same parameters, but it start at an altitude $h = 7,000$ [m]. Control deflections are bounded by 30 [deg] and exhibit a very minimal amount of overshoots. A smooth trajectory tracking is observed. The steady-state tracking error is nearly zero at 2.4 sec. The maximum value of the angle of attack α is less than 34.87 [deg]. The simulated responses in Fig. 16 show that this controller is sufficiently robust and maintain stability in the presence of turbulence and partial loss of control effective-ness due to damage in control surfaces. The advantage of the adaptive version over the classical one yielding smoother control signals was also seen in the simulated waveforms.

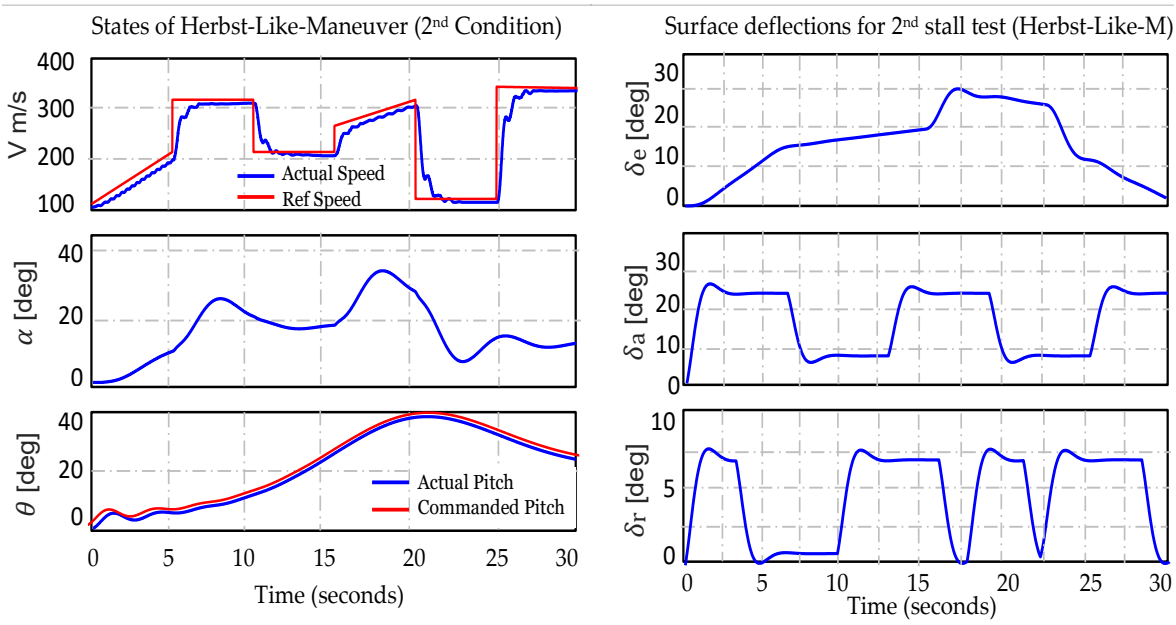

Fig. 16. States and control surface deflections for Herbst-Like-Maneuver (under the 2nd stall test)

Now we can summaries the above study in the Table 5 as shown.

Table 5. Adaptive symmetric control with Herbst-Like-Maneuver (under two test seniors)

Parameter	1st senior	2nd senior	Notes
Initial angle of attack (α_0) and Initial velocity (V_0)	1.5° 100 m/s	3.2° 107 m/s	Given as initial condition
Uncertainty in model (Δ)	±20%	±20%	Variations around nominal values
Angle of attack variation (α)	≤ 18° (high)	≤ 34.87° (high)	Ensured by neural controller
Pitch angle tracking (θ)	$e_\theta < 0.013$	$e_\theta < 0.025$	Effective (High precision) tracking
Sideslip angle deviation (β)	≤ 0.08°	≤ 0.05°	Maintained within limits
Actual speed of aircraft (V)	$\max\|\|V - V_{ref}\|\|_2 < 0.001$ m/s		Good tracking of V_{ref} (Precision)
Deflections of surfaces (δ)	$\delta \in [-30° \; 30°]$	$\delta \in [-25° \; 25°]$	Less efforts with little of variation
Performance Metrics for Different seniors in Fighter Aircraft Maneuvers			
Post-Stall Recovery Time (t_{ps})	3.451 s	4.132 s	We deduce that: adaptive symmetric control focuses on balanced maneuverability and stability, mainly used for standard evasive maneuvers. (i.e., 1m/s ≈ 1.94knots).
Response Time (t_r)	0.753 s	0.987 s	
Lift Coefficient (C_L)	Moderate (0.8 to1.2)		
Drag Coefficient (C_D)	Moderate (0.02 to0.04)		
Lift-to-Drag Ratio (L/D)	9.22		

The angle of attack variation remains within 18° and 34.87°, ensuring controlled high-AoA maneuvers while maintaining stability. Pitch angle tracking is highly precise, with tracking errors below 0.013 and 0.025, ensuring accurate trajectory following. Sideslip angle deviation is kept minimal (≤ 0.08° and ≤ 0.05°), reducing lateral instability. Control surface deflections range from ±30° to ±25°, indicating less control effort with minimal variation, optimizing efficiency. Post-stall

recovery time increases slightly (3.451 s to 4.132 s), suggesting a trade-off between maneuverability and stability. Response time remains moderate (0.753 s to 0.987 s), ensuring a balanced reaction time suitable for standard evasive maneuvers. Lift-to-drag ratio of 9.22 indicates efficient aerodynamic performance, supporting smooth maneuver execution. Adaptive symmetric control is optimized for balanced maneuverability and stability. It is mainly suited for standard evasive maneuvers, where stability and controlled response are crucial. The small high-frequency activity observed in Fig. 15 and Fig. 16 is primarily attributed to the sliding-mode term of the guidance component, which enhances convergence during rapid attitude transitions. However, the amplitude of these oscillations remains well within actuator tolerance limits and does not induce chattering or control saturation. In a real-time implementation, such high-frequency components could be further mitigated by a low-pass or boundary-layer filter in the sliding manifold, ensuring smooth actuator motion without compromising dynamic responsiveness.

6.2. Simulations of Adaptive Asymmetric Control

Without thrust vectoring, an aircraft is more likely to enter uncontrolled flight regimes, such as deep stalls or unrecoverable spins, during post-stall maneuvers. Purely aerodynamic control relies heavily on managing energy in terms of speed and altitude. Post-stall maneuvers require rapid changes in energy, which are much harder to achieve without thrust vectoring. Performing these maneuvers using only aerodynamic control surfaces is challenging and is largely constrained by the aircraft's design and aerodynamic capabilities. In this context, we demonstrate the effectiveness of the proposed control method during post-stall maneuvers. The controller design for such aggressive maneuvers, particularly at high angles of attack α, focuses on controlling the aircraft's attitude and angular velocity. Before initiating the maneuvers, we assume that 25% dynamic uncertainties and disturbances are present in the model. Additionally, it is assumed that the aircraft begins the maneuver from a wings-level steady flight at $h = 5$ [km], traveling at a speed of 0.88 Mach. The initial conditions are: $\alpha_0 = 0.70$ [deg], $\delta_{EL0} = \delta_{ER0} = -1.29$ [deg], thrust $\mathrm{F_{TH0}} = 22.860$ [N], throttle position $\delta_{\mathrm{TH0}} = 0.54$.

- Pull-Up Maneuver for Stall Testing: This maneuver is performed using the elevator to starts a climb from the initial altitude. The time histories of speed, angle of attack, and pitch angle for this maneuver are displayed in Fig. 17. Based on the initial conditions, the simulation indicates that the aircraft enters a stall after the 10^{th} second, remains in the post-stall region until the 26^{th} second, and then transitions into the deep stall region. In the deep stall region, despite continuous elevator commands to pull the aircraft up, an undesired nose-down pitching motion develops rapidly. The results suggest that the proposed aerodynamic control interventions should be applied no later than just before entering the deep stall region.

Recovering from a deep stall using only aerodynamic controls involves reducing the angle of attack by pushing the nose down, managing elevons appropriately, using the rudder to maintain directional control, and avoiding aileron inputs. The key is to regain effective airflow over the wings and reestablish normal flight conditions as quickly as possible. For the Su-30, recovering from aggressive pull-up maneuvers using aerodynamic controls alone is highly difficult but not entirely impossible, depending on the severity of the maneuver and the flight regime. The Su-30 is designed with advanced aerodynamics and powerful control surfaces, but it also relies on thrust vectoring to enhance its maneuverability, especially in post-stall or deep stall conditions.

- Cobra Maneuver: was first demonstrated by the Russian pilot Pougachev, the Cobra maneuver involves two distinct phases. The first phase, known as the pull-up phase, entails the pilot executing a nose-up movement until the aircraft stalls and experiences a significant reduction in speed. The second phase, the recovery phase, involves transitioning the nose-down maneuver to bring the aircraft back to the intended pitch angle. During the pull-up phase, the maneuver is controlled by the pilot, while the recovery phase can be managed either through aerodynamic controls alone or by employing thrust vector control (TVC). In simulations, the pull-up phase begins under specific initial conditions, causing the aircraft to

ascend from its starting altitude. After 12 [s], when the aircraft reaches a stall condition with an angle of attack $\alpha = 20$ [deg] and a pitch angle $\theta = 73$ [deg], the recovery phase is initiated using only aerodynamic controls. This phase aims to adjust the pitch angle to a target of 10 [deg] within 14 [s]. Results indicate that if recovery begins when α exceeds 23 [deg], achieving the desired maneuver without elevator deflection saturation becomes impractical. Therefore, to ensure the maneuver is completed with aerodynamic controls alone (no thrust vector control), the recovery phase must commence no later than when α reaches $\alpha = 23$ [deg]. The time histories for speed, angle of attack, and pitch angle are illustrated in Fig. 18 for a successful recovery phase achieved with aerodynamic control only.

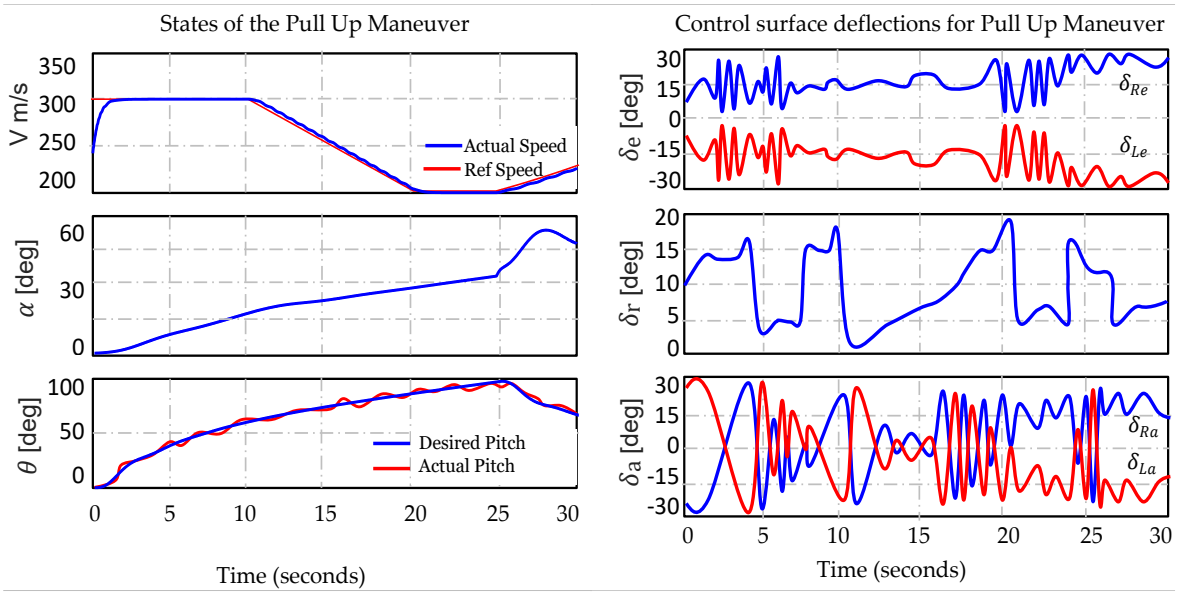

Fig. 17. States and control surface deflections for pull up-maneuver (aerodynamic control)

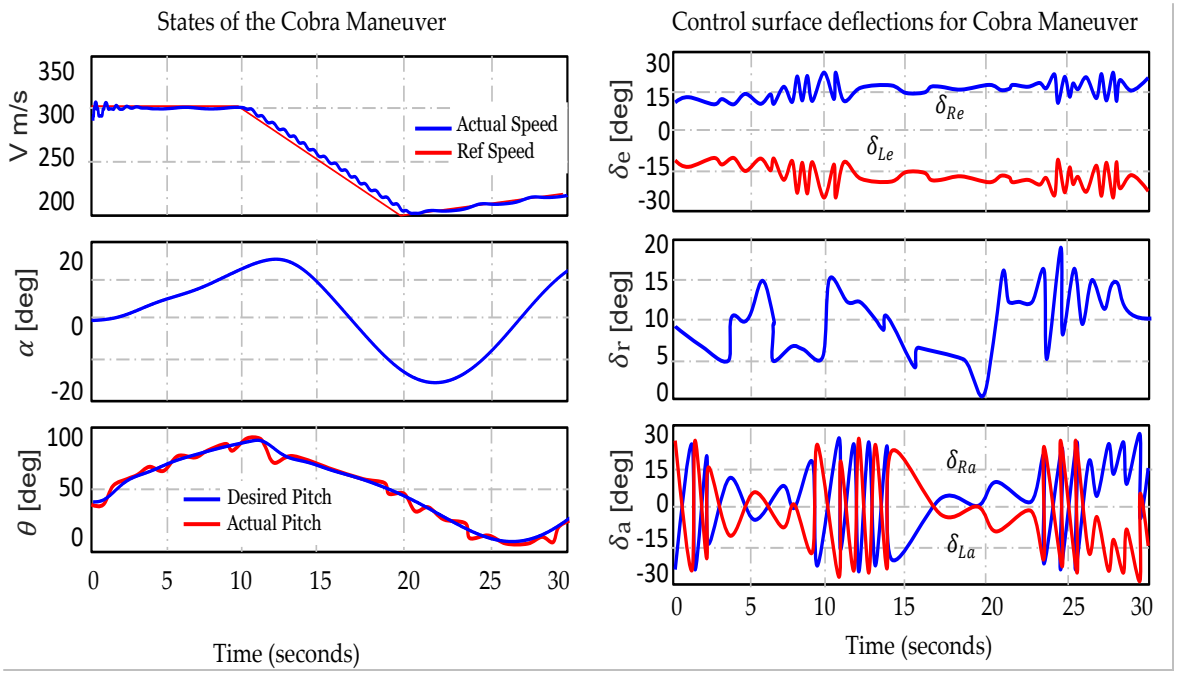

Fig. 18. States and control surface deflections for Cobra-Maneuver (aerodynamic control)

The findings are summarized in Table 6.

Table 6. Adaptive asymmetric control (pull-up-maneuver vs cobra-maneuver)

Parameter	Cobra	Pull-Up	Notes
Initial angle of attack (α_0) and Initial velocity (V_0)	0.70° 213 m/s	0.30° 295 m/s	Given as initial condition
Uncertainty in model (Δ)	±25%	±25%	Variations around nominal values
Angle of attack variation (α)	≤ 57.82° very high	≤ 19.23° (high)	Pull-Up maneuver requires high AoA to generate lift, while Cobra maneuver relies on controlled pitch with lower AoA
Pitch angle tracking (θ)	$\max_\theta \leq 100°$	$\max_\theta \leq 95.7°$	Cobra maneuver requires precise pitch control to maintain stability, while Pull-Up involves rapid pitch increase
Sideslip angle deviation (β)	≤ 0.06°	≤ 0.07°	Maintained within limits
Actual speed of aircraft (V)	$\max\|V - V_{\text{ref}}\|_2 < 0.000952$ m /s		Good tracking ensured by NN controller
Deflections of surfaces (δ)	$\delta \in [-29° \;\; 29°]$	$\delta \in [-24° \;\; 24°]$	Less efforts (significant smooth variation)
Performance Metrics for Different seniors in Fighter Aircraft Maneuvers			
Post-Stall Recovery Time (t_{ps})	2.512 s	3.125 s	It can be deduced that: Adaptive asymmetric control is designed for more aggressive maneuvers like Herbst-like and Spin maneuvers, leading to higher AoA and response rates.
Response Time (t_r)	0.652 s	0.831 s	
Lift Coefficient (C_L)	High (1.8 - 2.1 and 1.4 - 1.7)		
Drag Coefficient (C_D)	High (0.08 - 0.12 and 0.05 - 0.09)		
Lift-to-Drag Ratio (L/D)	8.54		

Performing the Cobra maneuver using only aerodynamic control surfaces (i.e., without thrust vector control, TVC) is theoretically possible but highly difficult and generally impractical. The maneuver requires pitching the aircraft to extreme angles of attack (AoA), beyond the range where conventional controls remain effective. Aircraft such as the Su-27, Su-30, and F-18, with advanced aerodynamics, can achieve Cobra-like maneuvers without TVC, though it is risky for most platforms. The Su-30, however, is specifically designed for such conditions, making the Cobra feasible and relatively manageable. Its design reduces induced drag and enhances pitch stability, allowing faster response and lower control effort, which highlights the efficiency of the adaptive asymmetric control in post-stall regimes. This study focuses on validating the proposed architecture under demanding conditions, while future work may compare it with baseline controllers (PID, NDI, reinforcement learning), though such benchmarking was limited here by the custom Su-30 dynamics and simulator constraints.

6.3. Simulations of Practical Adaptive Control

This subsection presents simulation results for the Su-30 dynamic model of (33) using the practical adaptive controller from Section 5. The state vector is initialized to zero except for the angle of attack α_0 and speed V_0, with the aircraft starting from $\mathbf{x}_0$. Uncertainties from un-modeled dynamics and disturbances are introduced as ±30% variations around nominal values. Several maneuvers were simulated with the proposed algorithm, and the results are shown in Fig. 19.

- Herbst-like maneuver: The target moves along the path:

$$[x_T(t) \;\; y_T(t) \;\; z_T(t)] = \left[x_{\max}\sin(w_T t)\cos\big(\theta(t)\big) \quad y_{\max}\sin(w_T t)\sin\big(\theta(t)\big) \quad z_{\max}\cos(w_T t)\right]$$

with $\theta(t) = \theta_{\max}\sin(w_T t)$, $\theta_{\max} = \pi/2$, and $w_T = \pi/30$.

- Spin maneuver: The target follows the equations: $\mathrm{r}_T(t+dt) = \mathrm{r}_T(t) + \mathrm{v}_T(t)dt$ where the velocity vector is $\mathrm{v}_T = [-A_T\cos(wt)/w; \;\; -304.8; \;\; A_T\sin(wt)/w]$, $[\mathrm{m/s}]$ and $\mathrm{a}_T(t)$ is: $\mathrm{a}_T(t) = A_T[\sin(wt) \;\; 0 \;\; \cos(wt)]$, $[\mathrm{m/s^2}]$.

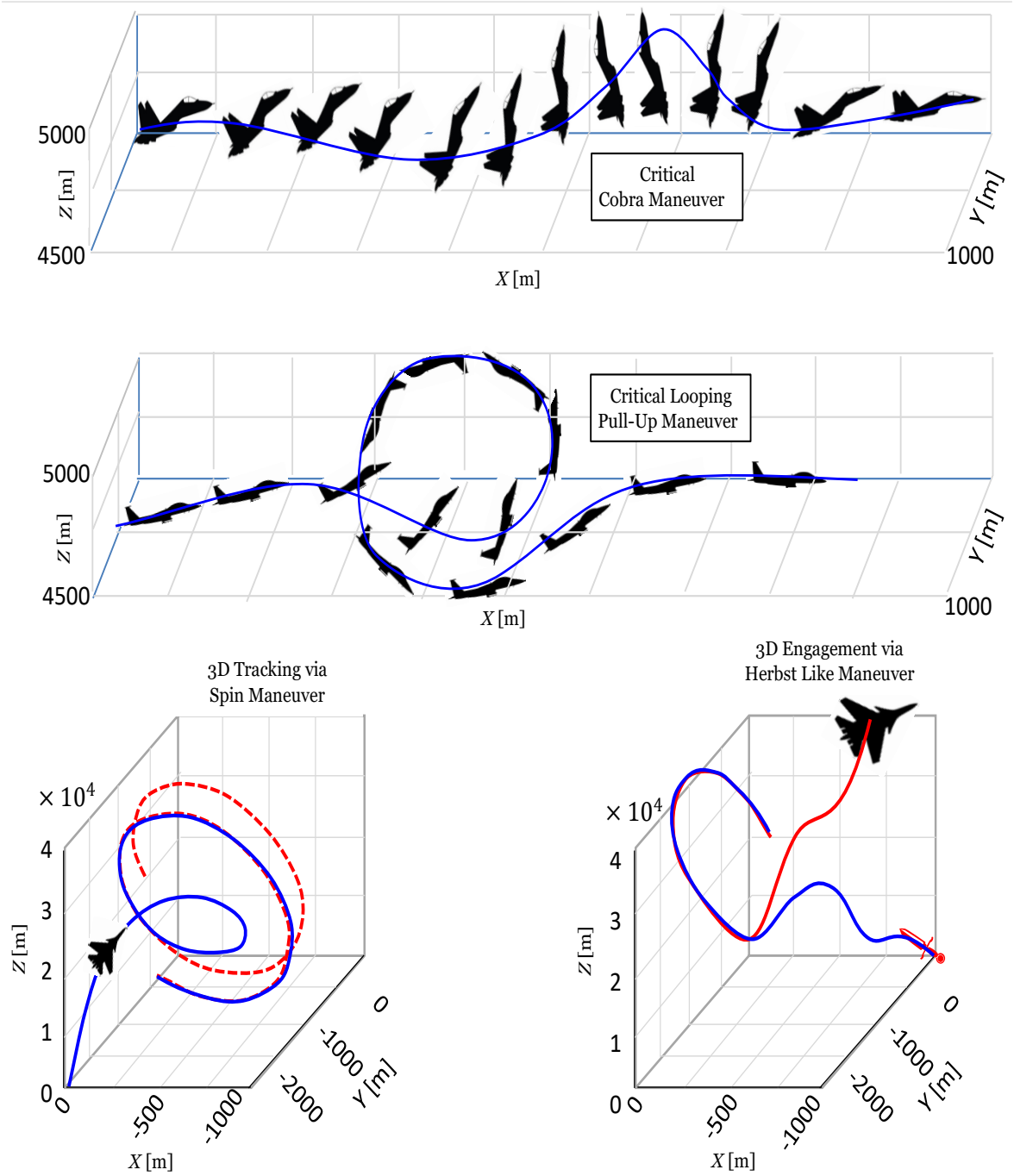

Fig. 19. Motion and flight trajectory visualization (3D engagement by practical adaptive control)

Pursuit of Herbst-like and Spin Maneuvers: The integration of the adaptive RBF neural controller with a pure pursuit guidance law, operating in parallel with the system's inverse dynamics, significantly enhances tracking precision and accuracy. The implementation of the Herbst-like and spin maneuvers leverages the proposed feedback-error-learning approach along with the derived stable tuning rule. The results of such control are shown in Fig. 20.

Fig. 20 illustrate the good tracking of the three dimensional evasive maneuvers by using the proposed neuro flight controller. By tuning (optimizing) the controller parameters, the system achieves excellent tracking of both the high angle of attack α and roll angle ϕ, with deviations in sideslip angle β kept below 0.3^{o}, ensuring effective tracking. The control inputs remain within actuator limits, with deflections not exceeding $|\delta_a| \leq 20^{o}$, $|\delta_r| \leq 30^{o}$, and $|\delta_e| \leq 25^{o}$,

demonstrating that saturation is avoided. These simulation results, as shown in Fig. 20 based on a high-performance Su-30 fighter aircraft model, highlight that the developed control strategy and tuning rule facilitate a more efficient and higher-performing control network. Now we can summaries the above study in the Table 7.

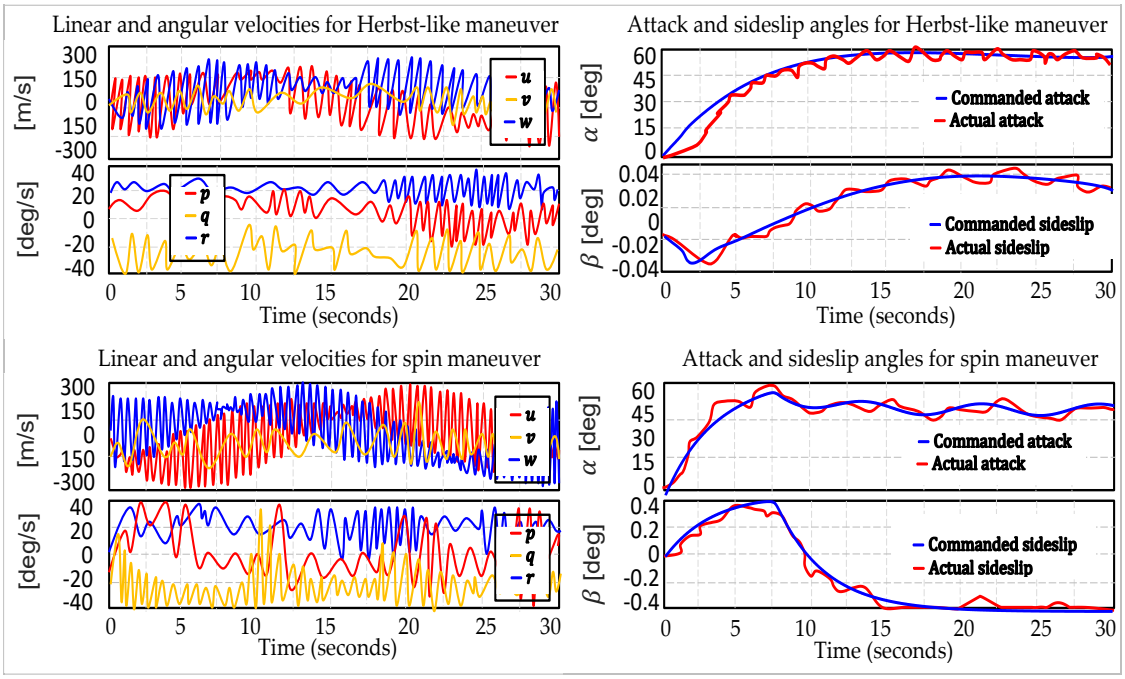

Fig. 20. Time histories of velocities, α and β for the practical adaptive control

Table 7. Practical adaptive symmetric control (looping-pull-up/herbst-like maneuvers)

Parameter	Looping (Spin)	Herbst-like	Notes
Initial angle of attack (α_0) and initial velocity (V_0)	0.12° 250 m/s	0.05° 270 m/s	Given as initial condition
Uncertainty in model (Δ)	±30%	%30±	Variations around nominal values
Angle of attack variation (α)	≤ 53.52° Very high	≤ 59. 3° (Very high)	Indicates extreme maneuverability with high instability risk
Sideslip angle deviation (β)	≤ 0.38°	≤ 0.04°	Herbst-like maneuver reduces undesired F_Y forces more than spin maneuver.
Actual speed of aircraft (V)	$\|V - V_{\text{ref}}\| \in [0.85\ \ 0.93] \times 10^{-3}$		Good tracking is ensured
Deflections of surfaces (δ)	$\delta \in [-28°\ 28°]$	$\delta \in [-26°\ 26°]$	Less efforts with very sharp of variation
Performance Metrics for Different seniors in Fighter Aircraft Maneuvers			
Post-Stall Recovery Time t_{ps}	3.012 s	1.853 s	The practical adaptive control is optimized for extreme maneuverability, with the highest AoA and agility but at the cost of increased control surface usage and fuel consumption.
Response Time (t_r)	0.758 s	0.667 s	
Lift Coefficient (C_L)	Very high (1.85 to 1.92)		
Drag Coefficient (C_D)	Very high (0.32 to 0.38)		
Lift-to-Drag Ratio (L/D)	7.12 to 7.56		

7. Conclusion and Directions for Future Work

This paper presents an advanced adaptive control framework designed to enhance the performance, robustness, and autonomy of the Su-30 fighter aircraft under dynamic and uncertain flight conditions. Three complementary control strategies are developed and analyzed: (i) an adaptive symmetric-surface controller for precise speed and trajectory tracking across varying Mach numbers and altitudes; (ii) an adaptive asymmetric-surface controller enabling stable post-stall maneuvers without thrust vectoring, ensuring reliable handling at high angles of attack where classical methods often fail; and (iii) a practical adaptive control scheme integrating nonlinear dynamic inversion, RBF neural networks, and a 3D sliding pure-pursuit guidance law for coordinated maneuvering and target tracking. These strategies collectively address key challenges in under-actuated dynamics and unmodeled aerodynamic effects, achieving robust performance during complex maneuvers such as Cobra, Herbst-like, and spin. The RBFNN adaptation improves responsiveness to aerodynamic uncertainty, while the integrated guidance–control architecture ensures accurate trajectory shaping under extreme conditions. High-fidelity simulations confirm the framework's effectiveness and real-time feasibility within advanced software environments. The architecture is theoretically sound and implementable in high-fidelity simulators, with future work focusing on Hardware-in-the-Loop validation and the inclusion of thrust vectoring and sensor effects to further extend realism and applicability. This study pioneers the integration of asymmetric neural-adaptive control with a sliding-mode-enhanced pursuit guidance law within a unified NDI framework for fourth-generation fighter aircraft. Unlike existing NDI or neural-augmented designs limited to symmetric configurations, the proposed architecture enables stable post-stall and high-angle-of-attack maneuvers previously unachievable with conventional autopilots. The approach demonstrates how combining adaptive learning, nonlinear inversion, and hybrid control allocation can significantly extend maneuverability and reliability, establishing a transferable framework for future intelligent flight-control systems.

Future work will focus on extending the study toward higher-fidelity and experimentally validated implementations. Planned developments include the incorporation of thrust vectoring effects, actuator dynamics, and sensor noise to improve model realism and robustness evaluation. The computational aspects of real-time execution will be formally analyzed, including delay and discretization effects, to assess the controller performance under embedded hardware constraints. To mitigate high-frequency control activity, filtering strategies, adaptive gain scheduling, and rate limiting will be investigated to ensure smooth actuation without compromising responsiveness. In addition, comparative analyses with alternative estimation techniques such as recursive least squares, TLSE, and Kalman-based filters will be carried out under variable excitation and noise conditions. Finally, benchmarking the proposed adaptive framework against conventional and modern control methods, including classical NDI, PID, and reinforcement learning approaches, will provide a more comprehensive assessment of tracking accuracy, robustness, and computational efficiency in realistic flight scenarios.

Author Contribution: All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix

A representative formulation of the aerodynamic force and moment coefficients, expressed in the body-fixed reference frame, is given by:

$$\begin{cases} C_x = C_{x0} + C_{x1}\alpha + C_{x2}\alpha^2 + C_{x3}\dfrac{qc}{V} + C_{x4}\delta_e + C_{x5}\delta_r + C_{x6}\delta_a + C_{x7}\mathrm{F_T} \\ C_y = C_{y0} + C_{y1}\beta + C_{y2}\beta^2 + C_{y3}\dfrac{pb}{2V} + C_{y4}\dfrac{rb}{2V} + C_{y5}\delta_e + C_{y6}\delta_r + C_{y7}\delta_a + C_{x8}\beta\mathrm{F_T} \\ C_z = C_{z0} + C_{z1}\alpha + C_{z2}\alpha^2 + C_{z3}\dfrac{qc}{V} + C_{z4}\delta_e + C_{z5}\delta_r + C_{z6}\delta_a + C_{z7}\alpha F_T \\ C_l = C_{l0} + C_{l1}\beta + C_{l2}\dfrac{pb}{2V} + C_{l3}\dfrac{rb}{2V} + C_{l4}\delta_e + C_{l5}\delta_r + C_{l6}\delta_a + C_{l7}(y_{\mathrm{gc}} - y_{\mathrm{s}})\mathrm{F_T} \\ C_m = C_{m0} + C_{m1}\alpha + C_{m2}\dfrac{qb}{2V} + C_{m3}\dfrac{\dot{\alpha}c}{2V} + C_{m4}\delta_e + C_{m5}\delta_r + C_{m6}\delta_a + C_{m7}\alpha\mathrm{F_T} \\ C_n = C_{n0} + C_{n1}\beta + C_{n2}\dfrac{pb}{2V} + C_{n3}\dfrac{rb}{2V} + C_{n4}\delta_e + C_{n5}\delta_r + C_{n6}\delta_a + C_{n7}(z_{\mathrm{gc}} - z_{\mathrm{s}})\mathrm{F_T} \end{cases}$$

with x_{gc}, y_{gc} and z_{gc} are positions of the center of gravity and x_{s}, y_{s}, and z_{s} are positions of the specific force sensors. These aerodynamic equations can be represented by $Y = A\Theta$ or $[A \mid Y][\Theta^\top \mid -1]^\top = 0$ where $Y = [C_x,\ C_y,\ C_z,\ C_l,\ C_m,\ C_n]^\top \in \Re^{6\times 1}$ is the vector of the variable need to be filled from experimental data, $A = [A_1, \dots A_6]^\top \in \Re^{6\times p}$ is the dimensional matrix of explanatory variables, and $\Theta = [\Theta_1, \dots, \Theta_p]^\top \in \Re^{p\times 1}$ is the vector of system parameters.

$$\begin{cases} A = \begin{bmatrix} 0_{1\times 8} & 1 & \alpha & \alpha^2 & qc/V & & & \delta_e & \delta_r & \delta_a & \mathrm{F_T} & 0_{1\times 41} \\ 0_{1\times 17} & 1 & \beta & \beta^2 & pb/2V & rb/2V & & \delta_e & \delta_r & \delta_a & \beta\mathrm{F_T} & 0_{1\times 32} \\ 0_{1\times 25} & 1 & \alpha & \alpha^2 & qc/V & & & \delta_e & \delta_r & \delta_a & \alpha\mathrm{F_T} & 0_{1\times 24} \\ 0_{1\times 33} & 1 & \beta & & pb/2V & rb/2V & & \delta_e & \delta_r & \delta_a & (y_{\mathrm{gc}} - y_{\mathrm{s}})\mathrm{F_T} & 0_{1\times 16} \\ 0_{1\times 41} & 1 & \alpha & & qb/2V & \dot{\alpha}c/2V & & \delta_e & \delta_r & \delta_a & \alpha\mathrm{F_T} & 0_{1\times 8} \\ & 1 & \beta & & pb/2V & rb/2V & & \delta_e & \delta_r & \delta_a & (z_{\mathrm{gc}} - z_{\mathrm{s}})\mathrm{F_T} & \end{bmatrix} \in \Re^{6\times p};\ Y = \begin{bmatrix} C_x \\ C_y \\ C_z \\ C_l \\ C_m \\ C_n \end{bmatrix} \in \Re^{6\times 1} \\ \Theta = [C_{x0}, \dots, C_{x7} \ \vdots\ C_{y0}, \dots, C_{y8} \ \vdots\ C_{z0}, \dots, C_{z7} \ \vdots\ C_{l0}, \dots, C_{l7} \ \vdots\ C_{m0}, \dots, C_{m7} \ \vdots\ C_{n0}, \dots, C_{n7}]^\top \in \Re^{p\times 1} \end{cases}$$

Both the response vector Y and specific columns of the explanatory variable matrix A originate from measurements that are inherently affected by noise and uncertainty. In such cases, it is essential to distinguish clearly between the true underlying values and the observed (measured) data. Since these coefficients can vary due to changes in speed, altitude, angle of attack, and other factors, averaging $\Theta^{\mathrm{ref}} = \mathbb{E}[\Theta(k)]$ helps in developing control laws and simulations that are more robust and less sensitive to minor fluctuations. This approach is especially useful when precise, real-time data is unavailable or when a general, steady-state performance prediction is needed.

Author Contribution: All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] N. Wang, R. Ma, X. Chang and L. Zhang, "Numerical Virtual Flight Simulation of Quasi-Cobra Maneuver of a Fighter Aircraft," *Journal of aircraft*, 2020, https://doi.org/10.2514/1.C035687.

[2] Y. Li, L. Zhang, C. Gao, J. Zhu, B. Dong, "Investigation of Asymmetric Flow of a Slender Body with Low-Aspect Ratio Fins Having Large Deflection Angle," *Aerospace*, vol. 11, no. 10, p. 835, 2024, https://doi.org/10.3390/aerospace11100835.

[3] X. Gao, Y. Zhang, B. Wang, Z. Leng, and Z. Hou, "The Optimal Strategies of Maneuver Decision in Air Combat of UCAV Based on the Improved TD3 Algorithm," *Drones*, vol. 8, no. 9, p. 501, 2024, https://doi.org/10.3390/drones8090501.

[4] E. R. Van Oort, L. Sonneveldt, Q. P. Chu, and J. A. Mulder, "Full-envelope modular adaptive control of a fighter aircraft using orthogonal least squares," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 5, pp. 1461-1472, 2010, https://doi.org/10.2514/1.48175.

[5] W. B. Herbst, "Dynamics of air combat," *Journal of Aircraft*, vol. 20, no. 7, pp. 594-598, 1983, https://doi.org/10.2514/3.44916.

[6] D. Canter and A. Groves, "X-31 post-stall envelope expansion and tactical utility testing," *Biennial Flight Test Conference*, p. 2171, 1994, https://doi.org/10.2514/6.1994-2171.

[7] Y. Li, N. Sundararajan, and P. Saratchandran, "Neuro-controller design for nonlinear fighter aircraft maneuver using fully tuned RBF networks," *Automatica*, vol. 37, no. 8, pp. 1293-1301, 2001, https://doi.org/10.1016/S0005-1098(01)00090-5.

[8] H. K. Khalil, "Nonlinear Systems," *Upper Saddle River*, 2002, https://testbankdeal.com/sample/nonlinear-systems-3rd-edition-khalil-solutions-manual.pdf.

[9] T. Lombaerts, G. Looye, J. Ellerbroek, and M. R. Y. Martin, "Design and piloted simulator evaluation of adaptive safe flight envelope protection algorithm," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 8, pp. 1902-1924, 2017, https://doi.org/10.2514/1.G002525.

[10] S. Irfan, L. Zhao, S. Ullah, U. Javaid, and J. Iqbal, "Differentiator-and observer-based feedback linearized advanced nonlinear control strategies for an unmanned aerial vehicle system," *Drones*, vol. 8, no. 10, p. 527, 2024, https://doi.org/10.3390/drones8100527.

[11] P. Bhardwaj, V. S. Akkinapalli, J. Zhang, S. Saboo, and F. Holzapfel, "Adaptive augmentation of incremental nonlinear dynamic inversion controller for an extended f-16 model," *AIAA Scitech 2019 Forum*, p. 1923, 2019, https://doi.org/10.2514/6.2019-1923.

[12] N. Sahbon, M. Jacewicz, P. Lichota, and K. Strzelecka, "Path-Following Control for Thrust-Vectored Hypersonic Aircraft," *Energies*, vol. 16, no. 5, p. 2501, 2023, https://doi.org/10.3390/en16052501.

[13] W. Han and S. Zhang, "Finite Time Convergence Incremental Nonlinear Dynamic Inversion Based Attitude Control for Flying - Wing Aircraft with Actuator Faults," *2019 CAA Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS)*, pp. 750-755, 2019, https://doi.org/10.1109/SAFEPROCESS45799.2019.9213409.

[14] S. Zhang, W. Han, and Y. Zhang, "Finite time convergence incremental nonlinear dynamic inversion-based attitude control for flying—wing aircraft with actuator faults," *Actuators*, vol. 9, no. 3, p. 70, 2020, https://doi.org/10.3390/act9030070.

[15] A. Ait, "Origins of the Algerian Su-30MKA," *Military Watch Magazine*, 2022, https://militarywatchmagazine.com/article/algerian-su30mka-africa-fighter-evaluate.

[16] V. M. Kazak, D. O. Shevchuk, N. A. Tymoshenko and L. V. Pomytkina, "Intelligent control system against aircraft's structural damage in the flight," *2016 4th International Conference on Methods and Systems of Navigation and Motion Control (MSNMC)*, pp. 232-236, 2016, https://doi.org/10.1109/MSNMC.2016.7783150.

[17] DCS Center, "Su-33 Flanker D Flight Manual," *Eagle Dynamics*, 2024, https://www.digitalcombatsimulator.com/en/downloads/documentation/dcs-su-33_flight_manual_en/.

[18] DCS center, "Su-27 Flanker Flight Manual," *Eagle Dynamics*, 2024, https://www.digitalcombatsimulator.com/upload/iblock/ed7/Su-27%20DCS%20Flaming%20Cliffs%20Flight%20Manual%20EN.pdf.

[19] R. Buckley, J. Pryor, and B. H. Greenshields, "Autonomous Maneuvering: A Defense Advantage for AFSOC Aircraft," *Defense Technical Information Center*, 2022, https://apps.dtic.mil/sti/pdfs/AD1184803.pdf.

[20] M. Niestroy, R. Luckner, and C. Doll, "Flight control systems for aircraft engaged in air wake surfing for efficiency," *AIAA Scitech 2020 Forum*, 2020, https://dx.doi.org/10.2514/6.2020-1003.

[21] J. Lake, "Su-27 Flanker-Sukhoi Super fighter," *Osprey Aerospace*, 1992, https://books.google.co.id/books/about/Su_27_Flanker.html?id=v5aoAAAACAAJ&redir_esc=y.

[22] C. Lyu and R. Zhan, "Global Analysis of Active Defense Technologies for Unmanned Aerial Vehicle," *IEEE Aerospace and Electronic Systems Magazine*, vol. 37, no. 1, pp. 6-31, 2022, https://doi.org/10.1109/MAES.2021.3115205.

[23] N. Rani, K. Jindal, R. Chikkara, and N. Malik, "Empowering Defense: Harnessing AI for Next-Generation Warfare," *Artificial Intelligence-Enabled Businesses: How to Develop Strategies for Innovation*, pp. 289-310, 2025, https://doi.org/10.1002/9781394168002.ch1.

[24] A. Agarwal, C. Mohanta, and S. N. Mehta, "Drone Technologies: State-of-the-Art, Challenges, and Future Scope," *Drone Technology: Future Trends and Practical Applications*, pp. 1-19, 2023, https://doi.org/10.1002/9781394168002.ch1.

[25] J. Lee and Y. Kim, "Neural network-based nonlinear dynamic inversion control of variable-span morphing aircraft," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 234, no. 10, pp. 1624-1637, 2020, https://doi.org/10.1177/0954410019846713.

[26] E. Safwat, W. Zhang, A. Mohsen, and M. Kassem, "Design and analysis of a robust UAV flight guidance and control system based on a modified nonlinear dynamic inversion," *Applied Sciences*, vol. 9, no. 17, p. 3600, 2019, https://doi.org/10.3390/app9173600.

[27] Q. He, Y. Tan, X. Liu, Q. Jia and J. Liu, "Reconfigurable Nonlinear Dynamic Inversion for Attitude Control of a Structurally Damaged Aircraft," *IEEE Access*, vol. 8, pp. 199931-199943, 2020, https://doi.org/10.1109/ACCESS.2020.3035436.

[28] S. Zhang, H. Zhang, and K. Ji, "Incremental nonlinear dynamic inversion attitude control for helicopter with actuator delay and saturation," *Aerospace*, vol. 10, no. 6, p. 521, 2023, https://doi.org/10.3390/aerospace10060521.

[29] M. A. Balint, "Advances in Flight Control Systems," *InTech*, 2011, https://doi.org/10.5772/587.

[30] N. Cho, Y. Kim, and S. Park, "Three-dimensional nonlinear path-following guidance law based on differential geometry," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 2503-2508, 2014, https://doi.org/10.3182/20140824-6-ZA-1003.00171.

[31] J. Ferrin, R. Leishman, R. Beard and T. McLain, "Differential flatness based control of a rotorcraft for aggressive maneuvers," *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2688-2693, 2011, https://doi.org/10.1109/IROS.2011.6095098.

[32] B. M. Safinaz, B. Mohammed, and B. Ahmed, "Observer Backstepping Design for Flight Control," *Engineering Proceedings*, vol. 29, no. 1, p. 3, 2023, https://doi.org/10.3390/engproc2023029003.

[33] S. N. Singh, M. L. Steinberg and A. B. Page, "Nonlinear adaptive and sliding mode flight path control of F/A-18 model," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1250-1262, 2003, https://doi.org/10.1109/TAES.2003.1261125.

[34] L. Sonneveldt, E. R. Van Oort, Q. P. Chu, and J. A. Mulder, "Nonlinear adaptive trajectory control applied to an F-16 model," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 1, pp. 25-39, 2009, https://doi.org/10.2514/1.38785.

[35] L. Cheng, Y. Li, J. Yuan, J. Ai, and Y. Dong, "L1 adaptive control based on dynamic inversion for morphing aircraft," *Aerospace*, vol. 10, no. 9, p. 786, 2023, https://doi.org/10.3390/aerospace10090786.

[36] J. Chen, J. Wang, and W. Wang, "Robust adaptive control for nonlinear aircraft system with uncertainties," *Applied Sciences*, vol. 10, no. 12, p. 4270, 2020, https://doi.org/10.3390/app10124270.

[37] I. D. Landau and B. Courtiol, "Design of multivariable adaptive model following control systems," *Automatica*, vol. 10, no. 5, pp. 483-494, 1974, https://doi.org/10.1016/0005-1098(74)90049-1.

[38] G. Dhiman, A. Y. Tiumentsev, and Y. V. Tiumentsev, "Neural Network and Hybrid Methods in Aircraft Modeling, Identification, and Control Problems," *Aerospace*, vol. 12, no. 1, p. 30, 2025, https://doi.org/10.3390/aerospace12010030.

[39] Z. Akhtar, S. A. Z. Naqvi, Y. A. Khan, M. T. Hamayun, and S. Ijaz, "Design and Experimental Validation of an Adaptive Multi-Layer Neural Network Observer-Based Fast Terminal Sliding Mode

Control for Quadrotor System," *Aerospace*, vol. 11, no. 10, p. 788, 2024, https://doi.org/10.3390/aerospace11100788.

[40] J. Boiffier, "The Dynamics of Flight: the equations," *Aircraft Engineering and Aerospace Technology: An International Journal*, vol. 71, no. 1, 1998, https://doi.org/10.1108/aeat.1999.12771aae.002.

[41] R. H. Barnard, "Mechanics of Flight," *Pearson Prentice Hall*, 2006, https://soaneemrana.com/onewebmedia/MECHANICS%20OF%20FLIGHT%20BY%20A.C%20KERMODE.pdf.

[42] D. Raymer, "Aircraft Design: A Conceptual Approach," *American Institute of Aeronautics and Astronautics*, 2012, https://doi.org/10.2514/4.104909.

[43] R. F. Stengel, "Flight Dynamics," *Princeton University Press*, 2004, https://doi.org/10.2307/j.ctt1287kgx.

[44] S. A. Emami and A. Banazadeh, "Intelligent trajectory tracking of an aircraft in the presence of internal and external disturbances," *International Journal of Robust and Nonlinear Control*, vol. 29, no. 16, pp. 5820-5844, 2019, https://doi.org/10.1002/rnc.4698.

[45] Z. Mohamed, A. Mohamed, F. Khaled, A. Souhila, and B. N. Sarah, "Application of the total least squares estimation method for an aircraft aerodynamic model identification," *International Journal of Computer and Systems Engineering*, vol. 10, no. 6, pp. 1152-1161, 2016, https://d1wqtxts1xzle7.cloudfront.net/47662420.

[46] Z. Mohammed, F. Khaled, "A Virtual Method to Estimate the Aerodynamic Coefficients of UAV," *Unmanned Systems*, vol. 10, no. 2, pp. 147-157, 2022, https://doi.org/10.1142/S230138502250008X.

[47] G. H. Golub and C. F. V. Loan, "Matrix Computations," *JHU Press*, 2013, https://books.google.co.id/books/about/Matrix_Computations.html?hl=de&id=X5YfsuCWpxMC&redir_esc=y.

[48] H. Liu and X. Gao, "Identification of nonlinear aerodynamic systems with application to transonic aeroelasticity of aircraft structures," *Nonlinear Dynamics*, vol. 100, no. 2, pp. 1037-1056, 2020, https://doi.org/10.1007/s11071-020-05553-2.

[49] M. Laban and J. A. Mulder, "On-line identification of aircraft aerodynamic model parameters," *IFAC Proceedings Volumes*, vol. 25, no. 15, pp. 199-204, 1992, https://doi.org/10.1016/S1474-6670(17)50633-3.

[50] A. Hamissi, K. Busawon, L. Dala, Y. Bouzid, M. Zaouche, and M. Hamerlain, "A new nonlinear control design strategy for fixed wing aircrafts piloting," *Unmanned Systems*, vol. 9, no. 04, pp. 293-308, 2021, https://doi.org/10.1142/S2301385021500126.

[51] J. Roskam, "Utilization of Separate Surface Control Systems on General Aviation Aircraft," *SAE Technical Paper*, 1977, https://doi.org/10.4271/770471.

[52] M. H. Sadraey, "Aircraft Design: A Systems Engineering Approach," *Wiley*, 2012, https://books.google.co.id/books?id=VT-Tc3Tx5aEC&printsec=copyright&redir_esc=y#v=onepage&q&f=false.

[53] G. Bujgoi and D. Sendrescu, "Tuning of pid controllers using reinforcement learning for nonlinear system control," *Processes*, vol. 13, no. 3, p. 735, 2025, https://doi.org/10.3390/pr13030735.

[54] A. Kalra, S. Anavatti, and R. Padhi, "Aggressive formation flying of fixed-wing UAVs with differential geometric guidance," *Unmanned Systems*, vol. 5, no. 02, pp. 97-113, 2017, https://doi.org/10.1142/S2301385017500078.

[55] H. B. Khamseh and F. Janabi-Sharifi, "Ukf–based lqr control of a manipulating unmanned aerial vehicle," *Unmanned Systems*, vol. 5, no. 03, pp. 131-139, 2017, https://doi.org/10.1142/S2301385017400015.

[56] H. Aygun, M. E. Cilgin, I. Ekmekci, and O. Turan, "Energy and performance optimization of an adaptive cycle engine for next generation combat aircraft," *Energy*, vol. 209, p. 118261, 2020, https://doi.org/10.1016/j.energy.2020.118261.

[57] B. Bekhiti, "Advanced Strategies for Guidance & Control of Surface-Air Missiles," *LAP Lambert Academic Publishing*, 2024, https://www.researchgate.net/publication/382125192_Advanced_Strategies_for_Guidance_Control_of_Surface-Air_Missiles.

[58] B. Bekhiti, G. F. Fragulis, M. Rahmouni, and K. Hariche, "A Novel Three-Dimensional Sliding Pursuit Guidance and Control of Surface-to-Air Missiles," *Technologies*, vol. 13, no. 5, p. 171, 2025, https://doi.org/10.3390/technologies13050171.

[59] P. Simplício, P. Acquatella, and S. Bennani, "Design and Analysis of a Launcher Flight Control System Based on Incremental Nonlinear Dynamic Inversion," *Aerospace*, vol. 12, no. 4, p. 296, 2025, https://doi.org/10.3390/aerospace12040296.

[60] A. Losekamm, "Fighter Jets for Ukraine: A Reality Check," *Romanian Military Thinking*, no. 4, pp. 320-333, 2023, https://doi.org/10.55535/RMT.2023.4.19.

[61] G. U. O. Yong, W. A. N. G. Changqing, and L. I. Aijun, "Robust formation control for missiles with obstacle avoidance," *Chinese Journal of Aeronautics*, vol. 35, no. 1, pp. 70-80, 2022, https://doi.org/10.1016/j.cja.2021.06.003.

[62] D. McLean, "Active control of a modern fighter aircraft," *Computer Control of Real-time Processes*, no. 41, p. 256, 1990, https://doi.org/10.1049/PBCE041E_ch15.

[63] P. Yang, B. Xiao, X. Chen, and L. Guo, "3D path planning problem for fighter aircraft with multiple constraints," *The Journal of Engineering*, vol. 2023, no. 11, p. e12325, 2023, https://doi.org/10.1049/tje2.12325.

[64] M. F. Manzoor, J. A. Paulson, Y. A. Butt, F. U. Rehman, and A. I. Bhatti, "Real-time implementation of nonlinear model predictive control for high angle of attack Maneuvers in fighter aircrafts using deep learning," *Systems Science & Control Engineering*, vol. 12, no. 1, p. 2342819, 2024, https://doi.org/10.1080/21642583.2024.2342819.